Perception multimodale de l'homme pour l'interaction Homme-Robot

Christophe Mollaret

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *03/12/2015* **par :**

Christophe Mollaret

Perception multimodale de l'homme pour l'interaction Homme-Robot

---

JURY

Laurent BESACIER	Professeur des Universités	Président du jury
Mohamed CHETOUANI	Professeur des Universités	Rapporteur
Etienne COLLE	Professeur des Universités	Rapporteur
Frédéric LERASLE	Professeur des Universités	Examinateur
Julien PINQUIER	Maître de Conférences	Examinateur
Isabelle FERRANE	Maître de Conférences	Examinateur

École doctorale et spécialité :
MITT : Signal, Image, Acoustique et Optimisation

Unité de Recherche :
Institut de Recherche en Informatique de Toulouse (UMR5505)

Directeur(s) de Thèse :
Frédéric LERASLE et *Isabelle FERRANÉ*

Rapporteurs :
Mohamed CHETOUANI et *Etienne COLLE*

Résumé de thèse

Cette thèse porte sur la perception multimodale de l'homme pour l'Interaction Homme-Robot (IHR). Elle a été financée par le projet ANR Contint RIDDLE (2012 – 2015). Ce projet est centré sur le développement d'un robot d'assistance pour les personnes âgées atteintes de troubles cognitifs légers. Celui-ci a pour but de répondre à un besoin grandissant d'aide à domicile envers les personnes âgées vivant seules. En effet, la population vieillissant de plus en plus, on estime qu'environ 33% des français auront plus de 60 ans en 2060. L'enjeu est donc de programmer un robot interactif (via ses capacités perceptuelles) capable d'apprendre la relation entre l'usager et un sous-ensemble d'objets du quotidien de ce dernier, soit des objets pertinents, présents ou possiblement égarés dans l'espace partagé du robot et de l'usager. Dans ce cadre, il reste de nombreux verrous à lever, notamment en termes de : (i) perception conjointe de l'homme et de son environnement, (ii) d'intégration sur un système robotisé, (iii) de validation par des scénarii mettant en jeu le robot et une personne âgée en interaction avec quelques objets usuels. La finalité du projet est de voir le robot répondre aux interrogations relatives à une dizaine d'objets courants (définis par une étude préliminaire sur une population qui se plaint de troubles cognitifs) par des actions appropriées. Par exemple, le robot signalera l'emplacement d'un objet en se déplaçant vers lui, en le saisissant ou en donnant des indications orales quant à sa position si l'objet n'est pas atteignable. Le projet RIDDLE est multipartenaire : il regroupe la société Magellium, le Gérontopôle de Toulouse, l'équipe MINC du LAAS-CNRS et l'entreprise Aldebaran Robotics dont le robot doit servir de plateforme pour les démonstrations finales. Cette thèse a été co-encadrée par Frédéric Lerasle et Isabelle Ferrané respectivement enseignants-chercheurs dans les équipes RAP du LAAS-CNRS et SAMoVA de l'IRIT-UPS.

Lors de ce projet, nous avons, en partenariat avec le gérontopôle, défini un scénario robotique regroupant trois phases principales. Une phase de *monitoring* où le robot se trouve loin de l'utilisateur et l'observe de sa position, en attente d'une demande d'interaction, une phase d'interaction proximale où le robot se trouve proche de l'utilisateur et interagit avec lui, et enfin la transition qui permet au robot de passer d'une phase à l'autre. Ce scénario est donc construit de manière à créer un robot d'interaction proactif mais non-intrusif. Le caractère non-intrusif est matérialisé par la phase de *monitoring*. La proactivité est, quant à elle, matérialisée par la création d'un détecteur d'intentionnalité permettant au robot de comprendre de manière non-verbale la volonté de l'utilisateur de communiquer avec lui.

Les contributions scientifiques de cette thèse recoupent divers aspects du projet : le scénario robotique, le détecteur d'intentionnalité, une technique de filtrage par essaim de particules, et enfin

une technique bayésienne d'amélioration du taux d'erreur de mot à partir d'informations de distance.

Cette thèse est divisée en quatre chapitres. Le premier traite du détecteur d'intentionnalité, le deuxième de la technique de filtrage développée, le troisième de la phase d'interaction proximale et des techniques employées, et enfin le dernier chapitre est centré sur les implémentations robotiques.

Remerciements

Je souhaiterais remercier mes deux laboratoires d'attache qui sont le LAAS-CNRS et l'IRIT-UPS pour l'acceuil dans leurs locaux et les moyens mis à ma disposition pour l'exécution de mes travaux. Je remercie également les équipes RAP et SAMoVA pour leur acceuil et soutien durant ces trois années.

Je tiens aussi à remercier Mohamed Chetouani et Etienne Colle pour le courage nécéssaire et le temps qu'ils ont consacré à la relecture et au rapport de mon manuscrit. Je les remercie également pour leur présence lors de ma soutenance. De même, je voudrais remercier mes examinateurs, Laurent Besacier et Julien Pinquier d'avoir fait partie de mon jury de thèse lors de ma soutenance.

Je me dois aussi de remercier Matthieu Herrb et Aurélie Clodic pour leur grande aide, et sans lesquels il m'aurait été impossible d'implémenter mes travaux sur les plateformes robotiques du LAAS-CNRS. J'aimerais aussi remercier l'ensemble de l'équipe MINC du LAAS-CNRS et en particulier Hervé Aubert pour notre collaboration amicale dans le projet RIDDLE. Je remercie également l'ensemble de mes camarades de thèse et stagiaires qui ont rendu ces trois ans plus doux. Je remercie ainsi François-Xavier, Dominique, Jean-Thomas, Lucas, Matthieu, Jérémy, Guilhem, François, Ali, Arthur, Maxime et Vincent bien qu'il soit difficile d'être exhaustif.

Je remercie aussi tout particulièrement Alhayat Ali Mekonen, post-doctorant au LAAS-CNRS, pour sa grande aide et son soutien durant ces trois ans de thèse, et sans qui celle-ci n'aurait pas eu la même portée.

Il convient aussi de remercier mes proches pour leur support, tout particulièrement Nadine Rouger pour son aide durant les dernières relectures.

Je remercie enfin grandement mes directeurs de thèse, Frédéric Lerasle et Isabelle Ferrané, pour leurs encouragements, leur aide, leurs conseils, leur expérience et leur patience durant ces trois ans, et sans qui tout ce travail n'aurait pas été possible.

Liste des publications associées à ces travaux

IEEE-ICASSP (International Conference on Acoustics, Speech and Signal Processing) 2016 (soumission) : A probabilstic scheme for fusion of multiple streams, multiple speech recognition systems. *Christophe Mollaret, Isabelle Ferrané, Julien Pinquier, Frédéric Lerasle*

IEEE-CVIU (Computer Vision and Image Understanding) 2016 (acceptée sous révisions) : A multi-modal perception based assistive robotic system for the elderly. *Christophe Mollaret, Alhayat Ali Mekonnen, Frédéric Lerasle, Isabelle Ferrané, Julien Pinquier, Blandine Boudet, Pierre Rumeau*

IEEE-ICME (International Conference on Multimedia and Expo) 2015 (Mollaret et al., 2015) : Perceiving user's intention-for-interaction : A probabilistic multimodal data fusion scheme. *Christophe Mollaret, Alhayat Ali Mekonnen, Isabelle Ferrané, Julien Pinquier, Frédéric Lerasle*

ACM (Association of Computing Machinery) : Audio Mostly 2014 (Pellegrini et al., 2014) : Towards soundpainting gesture recognition. *Thomas Pellegrini, Patrice Guyot, Baptiste Angles, Christophe Mollaret, Christophe Mangou*

IEEE-ICIP (International Conference on Image Processing) 2014 (Mollaret et al., 2014) : A particle swarm optimization inspired tracker applied to visual tracking. *Christophe Mollaret, Frédéric Lerasle, Isabelle Ferrané, Julien Pinquier*

Neurologie - Psychiatrie - Gériatrie 2014 (Boudet et al., 2014) : Quels sont les objets égarés à domicile par les personnes âgées fragiles ? Une étude pilote sur 60 personnes. *Blandine Boudet, Thérèse Giacobini, Isabelle Ferrané, Carine Fortin, Christophe Mollaret, Frédéric Lerasle, Pierre Rumeau*

33èmes journées annuelles de la société française de gériatrie et gérontologie 2013 (Boudet et al., 2013) : Étude pilote des objets recherchés sur une population de 60 personnes âgées ambulatoires. *Blandine Boudet, Carine Fortin, Thérèse Giacobini, Christophe Mollaret, Isabelle Ferrané, Frédéric Lerasle, Pierre Rumeau*

Glossaire

- **RGB :** Red, Green, Blue (rouge, vert, bleu)
- **RGB-D :** Red, Green, Blue, Depth (rouge, vert, bleu, profondeur)
- **RFID :** Radio Frequency Identification (radio-identification)
- **LED :** Light-Emitting Diode (diode électroluminescente)
- **SDK :** Software Development Kit (kit de développement logiciel)
- **API :** Application Programmation Interface (interface de programmation)
- **HRI :** Human-Robot Interaction (interaction homme-robot : HRI)
- **Genom :** Generator of Modules (générateur de modules)
- **ROS :** Robot Operating System
- **SVM :** Support Vector Machine (machine à vecteurs de support)
- **HMM :** Hidden Markov Model (modèle de Markov caché)
- **POMDP :** Partially Observable Markov Decision Processes
- **MLP :** MultiLayer Perceptrons (perceptron multi-couches)
- **DNN :** Deep Neural Network (réseau de neurones profond)
- **RNN :** Recurrent Neural Network (réseau de neurones récurant)
- **MLLR :** Maximum Likelihood Linear Regression (régression linéaire par maximum de vraisemblance)
- **GMM :** Gaussian Mixture Model (modèle de mélanges gaussiens)
- **MFCC :** Mel-Frequency Cepstrum Coefficients (coefficients cepstraux de fréquence Mel)
- **LPCC :** Linear Prediction Cepstral Coefficients (coefficients cepstraux de prédiction linéaire)
- **DCT :** Discrete Cosine Transform (transformée en cosinus discrète)
- **DFT :** Discrete Fourier Transform (transformée de Fourier discrète)
- **AAM :** Active Appearance Model (modèle d'apparence actif)
- **ICP :** Iterative Closest Point (point le plus proche itératif)
- **MCMC :** Méthode Combinatoire de Monte Carlo
- **PSO :** Particle Swarm Optimization (optimisation par essaim de particules)
- **SPSO :** Sequential Particle Swarm Optimization
- **PSOT :** Particle Swarm Optimization inspired Tracker (suivi par essaim de particules)
- **UKF :** Unscented Kalman Filter (filtre de Kalman sans parfum)
- **SIR :** Sampling Importance Resampling (échantillonnage avec rééchantillonnage par importance)
- **SIR_RW :** Sampling Importance Resampling with Random Walk model

– **SIR_CV :** Sampling Importance Resampling with Constant Velocity model
– **SLAM :** Simultaneous Localization And Mapping
– **WER :** Word Error Rate (taux d'erreur de mots)
– **WAcc :** Word Accuracy (précision de reconnaissance)
– **N-Bests :** N best utterances (N meilleurs hypothèses)
– **IG :** Information Gain (gain d'information)
– **TPR :** True Positive Rate
– **FAR :** False Alarm Rate
– **AED :** Average Early Detection
– **ACD :** Average Correct Duration
– **FPS :** Frame Per Second (image par seconde : IPS)
– N_{eff} **:** Nombre de particules efficaces
– **MAP :** Maximum A Posteriori
– **CMAP :** Conditional Maximum A Posteriori
– **MMSE :** Minimum Mean Square Error (minimum d'erreur quadratique moyenne)
– **SNR :** Signa to Noise Ratio (rapport signal sur bruit)
– **ZCR :** Zero-Crossing Rate (taux de passage par zéro)
– **RIDDLE :** Robot Interacting anD learning to help people in their Daily Life Environment
– **ANR :** Agence Nationale de la Recherche
– **LAAS :** Laboratoire d'Analyse et d'Architecture des Systèmes
– **CNRS :** Centre National de la Recherche Scientifique
– **IRIT :** Institut de Recherche en Informatique de Toulouse
– **UPS :** Université Paul Sabatier, Toulouse III
– **CMU :** Carnegie Mellon University
– **LIUM :** Laboratoire d'Informatique de l'Université du Maine
– **SAMoVA :** Structuration, Analyse et Modélisation de documents Vidéo et Audio
– **RAP :** Robotique, Action et Perception
– **MINC :** MIcro et Nanosystèmes pour les Communications sans fil
– **ADREAM :** Architectures Dynamiques Reconfigurables pour systèmes Embarqués Autonomes Mobiles
– **CHU :** Centre Hospitalier Universitaire
– **IEEE :** Institute of Electrical and Electronics Engineers
– **ICIP :** International Conference on Image Processing
– **ICME :** International Conference on Multimedia and Expo
– **ICASSP :** International Conference on Acoustics, Speech and Signal Processing
– **CVIU :** Computer Vision and Image Understanding
– **ACM :** Association of Computing Machinery

Table des matières

Introduction générale

Contexte et enjeux

Dans un contexte de vieillissement général des diverses populations du monde, le besoin d'assistance à domicile se fait de plus en plus grand. Malheureusement, ce besoin sociétal demeure insatisfait du fait d'un manque criant de personnel aidant, rendant l'assistance aux personnes âgées extrêmement coûteuse. Actuellement, l'assistance est le plus souvent réalisée par le compagnon (ou la compagne), les personnes seules ou veuves pouvant alors se retrouver dans une grande situation de faiblesse. D'ici 2060, on estime qu'environ 33% des français auront plus de 60 ans [1]. Cette tendance est donc amenée à s'intensifier, et les coûts d'assistance aux personnes âgées devraient augmenter. Une deuxième problématique concerne les personnes âgées atteintes de troubles cognitifs légers. Cette population est souvent touchée par de petites pertes de mémoire concernant les objets de son quotidien (lunettes, clés, etc.), ce qui peut occasionner des accusations de vol à l'encontre du personnel aidant (Boudet et al., 2014). En effet, le personnel médical relate régulièrement ce genre de situations qui engendrent une frustration chez l'un et l'impatience chez l'autre, la plupart des objets en question se révélant d'ailleurs simplement égarés ou mal rangés.

Le projet RIDDLE vient s'inscrire dans cette problématique de recherche d'objets perdus par un utilisateur atteint de troubles cognitifs légers. L'avantage de l'utilisation d'un robot dans une telle situation est qu'il ne pourra pas être soupçonné, un robot étant par nature incapable de vol. De plus, un robot ne s'impatientera pas de devoir répéter plusieurs fois par jour où se trouve un objet perdu. Enfin, un tel robot pourrait apporter un complément à l'assistance humaine, notamment lorsque l'aidant n'est pas présent au domicile de la personne âgée. Au-delà du projet RIDDLE (assistance à domicile pour la recherche d'objets), le robot pourrait alerter les secours en cas d'intrusion dans le domicile de son usager, ou bien encore appeler le SAMU en cas de chute ou de comportement anormal de l'utilisateur.

D'un point de vue scientifique, le projet RIDDLE est conçu comme un projet de perception et a pour objectif la programmation d'un robot d'assistance à la personne à son domicile. Ce projet se concentre sur la recherche d'une dizaine d'objets ciblés et sur l'interaction avec la personne afin de l'accompagner dans cette tâche. Pour cela, le robot doit non seulement percevoir l'environnement avec ses objets (comprendre où ceux-ci se trouvent dans la pièce), mais aussi être capable d'interagir avec la personne et mettre à jour les informations concernant l'état de l'environnement. Le robot

1. http ://www.insee.fr/

doit donc être équipé d'un certain nombre de capteurs extéroceptifs permettant une perception multimodale de l'environnement, soit aussi bien visuelle (position des objets, position de l'utilisateur, etc.) que sonore (interaction vocale avec l'utilisateur, bruits caractéristiques de l'environnement, etc.). En effet, dans notre contexte, l'environnement n'est équipé d'aucun capteur afin de faciliter l'installation du dispositif à domicile. Les capteurs sont tous embarqués sur le robot lui-même. Enfin, comme le robot se trouve dans la sphère privée de la personne, il se doit de rester discret et de s'adapter au mode de communication de son utilisateur de manière à ne pas être perçu comme une gêne par ce dernier. Selon cette logique de discrétion, il est nécessaire de déterminer une volonté de communication de la part de l'utilisateur. En effet, en observant l'interaction homme-homme, nous constatons que lorsqu'une personne veut discuter avec une autre, elle commence par se rapprocher, s'oriente vers elle, puis initie la discussion. Dans la robotique en revanche, la plupart des scénarii débutent directement par l'interaction sans prendre en compte ce processus préliminaire. Cela constitue donc un premier verrou perceptuel à prendre en compte dans nos scénarii.

Une autre difficulté réside dans la perception qu'a le robot de son utilisateur. L'être humain étant par nature non-statique, le robot doit posséder des capteurs suffisamment polyvalents pour lui permettre de percevoir l'homme dans toute la variabilité de son comportement (voix, déplacements, gestuelle, etc.). Malheureusement, la plupart des robots ne disposent pour l'instant que de peu de capteurs pour accomplir cette tâche. Lorsque l'utilisateur exécute des mouvements rapides, s'assoit ou tourne le dos au robot, les détecteurs en vision par ordinateur sont souvent mis en défaut. Des algorithmes de suivi peuvent combler ces lacunes. Le robot disposant de capacités de calcul limitées, il faut que ces algorithmes allient performance et légèreté en ressources CPU. Une autre solution peut être de fusionner les informations récoltées par d'autres capteurs ayant des champs d'action complémentaires. Un exemple typique est la fusion entre une image RGB (permettant souvent de percevoir un utilisateur à plus de 5m) et une image de profondeur (limitée à 4m de distance). Nous pouvons donc imaginer de nouveaux types de capteurs ou d'informations à fusionner.

Tous ces travaux ont été financés par l'ANR [2], sur le projet CONTINT RIDDLE [3] qui est pluri-disciplinaire. Le consortium est composé des sociétés Magellium et Aldebaran Robotics, s'occupant respectivement de la perception visuelle de l'environnement 3D, et de la plateforme robotique Roméo qui est ciblée pour l'implémentation à terme. Aldebaran participe aussi à la tâche de localisation visuelle des objets. Les équipes SAMoVA [4] de l'IRIT-UPS [5], et RAP [6] et MINC [7] du LAAS-CNRS [8] participent aussi au projet. L'IRIT est en charge de la partie interaction et représentation sémantique de l'environnement. L'équipe RAP est en charge de la partie perception de l'utilisateur, et enfin, MINC s'occupe de la radio-localisation d'objets par tags RFID. Enfin, dernier partenaire du projet, le gérontopôle de Toulouse apporte son expertise médicale en ce qui concerne la validation des scénarii robotiques et les expériences avec les personnes âgées.

2. Agence Nationale de la Recherche
3. Robot Interacting anD learning to help people in their Daily Life Environment
4. Structuration, Analyse et Modélisation de documents Vidéo et Audio
5. Institut de Recherche en Informatique de Toulouse, Université Paul Sabatier
6. Robotique, Action et Perception
7. MIcro et Nanosystèmes pour les Communications sans fil
8. Laboratoire d'Analyse et d'Architecture des Systèmes

Contributions scientifiques

Intentionnalité : notre première contribution scientifique traite de la perception d'intentionnalité. En effet, dans l'interaction homme-homme, une personne initiera une conversation en se tournant vers son interlocuteur sans nécessairement prononcer son nom. Ces travaux s'attachent donc à modéliser de manière multimodale ce comportement pour en tirer une information non-verbale que nous avons nommée « intentionnalité ». Celle-ci a été définie comme étant la volonté d'un utilisateur d'initier une session d'interaction avec le robot. Le détecteur d'intentionnalité se fonde sur des informations visuelles d'orientation de l'utilisateur, ainsi que sur les informations extraites du signal audio.

Filtrage : le filtrage est une problématique multiple et récurrente, particulièrement pour la perception qu'a le robot de son utilisateur. En effet ce dernier peut se déplacer et changer de posture très rapidement, compromettant les algorithmes de détection. Notre deuxième contribution scientifique se focalise sur un nouveau système de filtrage inspiré par l'algorithme d'optimisation par essaim de particules (Kennedy and Eberhart, 1995). Celui-ci est notamment utilisé pour améliorer la perception de l'utilisateur en ajoutant une cohérence temporelle aux diverses détections.

Fusion de systèmes combinés de reconnaissance vocale : notre troisième contribution scientifique est centrée sur l'amélioration de la reconnaissance vocale par la fusion de plusieurs microphones et moteurs de transcription. Étant donné la variabilité de notre contexte, il est très difficile de concevoir un système de reconnaissance vocale fiable. Nous proposons une architecture construite à partir de systèmes pré-existants afin d'améliorer les performances de reconnaissance globale, en prenant en compte la distance relative de l'utilisateur par rapport aux différents microphones.

Intégrations et démonstrations robotiques : ces travaux sont la suite logique des trois premières contributions. Le robot se trouvant au domicile du patient, il est primordial que celui-ci ne se sente pas envahi. Nous proposons donc un scénario robotique non-intrusif, démarrant par une étape de « pré-interaction » appelée *monitoring*. Le robot se sert ensuite du détecteur d'intentionnalité pour estimer de manière proactive une intention d'interaction et enclenche alors son déplacement vers l'utilisateur. La session d'interaction proximale peut alors s'ensuivre ou bien être avortée par l'utilisateur. Ces travaux sont complexes et chronophages car ils intègrent également les modalités développées par les partenaires du projet. Ce scénario permet toutefois de réaliser un couplage entre la perception de l'environnement et l'interaction homme-robot, ce qui a très rarement été rapporté dans la littérature scientifique.

Les contributions ainsi présentées se placent à différents niveaux d'abstraction. L'intentionnalité et le filtrage se situent dans une problématique de perception bas niveau (proche des capteurs), tandis que le problème de fusion et les scénarii robotiques se rapprochent d'un niveau d'abstraction symbolique. Nous avons ainsi pris en compte toute la chaîne d'abstraction, de l'exploitation directe des capteurs jusqu'à la conception du comportement du robot.

Plan du mémoire

Le mémoire est structuré en quatre chapitres.
Le premier chapitre concerne la détection d'intentionnalité, que nous définissons comme « l'intention de l'utilisateur de démarrer une phase d'interaction ». Les modalités audio-visuelles utilisées pour la construction du détecteur sont présentées avec les expériences et résultats associés.
Le deuxième chapitre se focalise sur la problématique de filtrage visuel. Nous présentons un filtre par essaim de particules directement inspiré de l'algorithme d'optimisation par essaim de particules, et exposons les résultats associés dans un contexte de suivi visuel. Ce filtre est notamment utilisé pour la perfectibilité de la détection d'intentionnalité.
Le troisième chapitre est quant à lui organisé autour de l'amélioration de l'interaction vocale. Nous présentons la série de modules embarqués sur le robot qui gèrent l'interaction avec l'utilisateur, ainsi qu'un dispositif de fusion de plusieurs microphones et moteurs de reconnaissance vocale qui permettent la diminution du WER (Word Error Rate, « taux d'erreur de mots » en français). D'autres modifications sont aussi envisagées au niveau du gestionnaire d'interaction.
Enfin, le dernier chapitre décrit le scénario robotique RIDDLE et les contributions scientifiques associées. Nous y présentons les implémentations réalisées sur différentes plateformes robotiques (PR2 et Nao) ainsi que les campagnes d'acquisitions et d'expérimentations réalisées à l'aide de ces robots dont nous avons tiré des conclusions d'observation et d'expériences. Chaque chapitre décrit les approches proposées et les expérimentations réalisées pour valider nos propositions.

Chapitre 1

Caractérisation de l'intentionnalité pour l'interaction homme-robot

1.1 Introduction

La robotique d'assistance pour personnes âgées est actuellement en vogue. Dans un contexte de vieillissement de la population globale, les métiers d'assistance à domicile sont en plein essor, quoiqu'ils demeurent très coûteux pour les familles, l'offre de personnel n'étant, pour l'heure, pas suffisante. L'installation d'un robot à domicile s'occupant des tâches répétitives peut donc constituer une solution pour soulager ou assister le personnel aidant. Nous nous intéressons ici au cas des personnes âgées atteintes de troubles cognitifs légers se manifestant, entre autres, par des pertes de mémoire à court terme. L'égarement d'objets de la vie de tous les jours, qui est la manifestation première des ces pertes de mémoire, peut conduire à des accusations de vols contre le personnel aidant (Boudet et al., 2014). Ceci engendre de la frustration aussi bien chez l'aidant, qui se sent injustement accusé, que chez la personne âgée qui ne retrouve pas ses objets. Le robot présente alors un double avantage : d'une part il est capable de surveiller un ensemble d'objets afin d'éviter leur égarement, et, d'autre part, il ne s'impatientera pas de répéter la même chose plusieurs fois dans une même journée.

Dans ce projet, nous ne voulons pas d'un système domotique qui engendrerait des travaux de modification du foyer de l'utilisateur. Un robot permet d'éviter cela puisqu'il ne nécessite *a priori* aucun aménagement particulier, si ce n'est une place de rangement lorsque celui-ci n'est pas en activité ou se recharge. Le contexte applicatif que nous avons mis en place se veut mono-utilisateur (une seule personne dans l'environnement sensoriel du robot). Toutes les expériences se déroulent dans une seule pièce de type environnement humain privatif (salon, cuisine, chambre, etc.). Le design du robot doit être acceptable, ainsi que son comportement, afin que l'utilisateur se sente à l'aise en sa présence et l'accepte. La non-intrusivité du robot est donc capitale car elle en déterminera l'acceptabilité. En effet, si celui-ci passe son temps à suivre l'utilisateur, son propriétaire risque fort de le trouver encombrant et envahissant dans son quotidien. C'est pour cela que le robot ne doit intervenir qu'à bon escient. Ainsi, dans notre contexte applicatif, le robot

commence par une phase de *monitoring* en scrutant l'utilisateur, puis tente de détecter son intention d'interaction. Il se déplace alors vers l'utilisateur de manière pro-active, c'est-à-dire sans demande explicite de celui-ci. Nous avons appelé cette volonté d'interagir avec le robot « intentionnalité de l'utilisateur ». Une fois celle-ci détectée, le robot entame une phase d'interaction proximale (proche de l'interlocuteur) avec la personne âgée, pour répondre à un éventuel besoin. Le projet s'est donc focalisé sur l'intégration d'un robot pro-actif mais non-intrusif. Dans ce chapitre, nous présentons un détecteur d'intentionnalité, permettant de passer d'une phase d'observation ou *monitoring* de l'utilisateur, à une phase d'interaction avec celui-ci.

Le détecteur d'intentionnalité doit donc être construit sur la base d'indices aussi bien verbaux que non-verbaux. Les informations non-verbales peuvent être décelées de deux façons :

- **par le contexte :** l'activité de l'utilisateur (regarder la télévision, parler au téléphone, etc.), le lieu d'interaction particulier (cuisine, salon, etc.) ou encore les variations du bruit ambiant (aspirateur, fenêtres ouvertes, etc.).
- **par le langage corporel :** l'orientation de l'utilisateur par rapport au robot, un mouvement de recul, une chute, mouvement des mains, etc.

Suivant cette définition, le détecteur d'intentionnalité a été conçu sur la base de trois modalités. La première prend en compte l'orientation du visage de l'utilisateur : plus celui-ci est orienté face au robot, plus la probabilité de vouloir interagir avec lui sera élevée. Suivant ce schéma, la deuxième modalité se base sur l'orientation des épaules. La troisième modalité concerne la détection d'activité vocale. Nous avons souhaité que le déclenchement de l'interaction ne soit pas limité à l'injonction de mots clef. L'intentionnalité doit pouvoir être détectée à une distance plus importante que celle *a priori* nécessaire à l'interaction proximale (inférieure à 1,50m). Pour cela, nous avons utilisé un détecteur d'activité vocale qui permet de détecter une zone de parole sans reconnaître ce qui a été prononcé. De plus, la détection d'activité vocale permet de repérer la parole quelle que soit la langue. Dans ce chapitre, nous présentons une architecture de détection d'intentionnalité d'une personne aux alentours du robot, schématisée sur la figure 1.1. L'algorithme utilise les images profondeur (Depth) et en couleurs (RGB) d'un capteur Kinect, ainsi que les buffers audio d'un microphone. Ces informations sont ensuite envoyées aux trois modules de détection déjà mentionnées : le détecteur d'orientation de visage, le détecteur d'orientation des épaules, et le détecteur d'activité vocale. Le tout est ensuite fusionné à l'aide d'un estimateur d'intentionnalité construit à partir d'un modèle de Markov caché (HMM).

Le chapitre est organisé de la façon suivante : tout d'abord, nous présentons dans la section 1.2 un état de l'art des différentes techniques utilisées pour détecter l'intentionnalité, en se focalisant sur les percepts employés pour notre détecteur. Les bilbiothèques ayant servi à l'implémentation sont décrites dans la section 1.3. Nous exposons ensuite les trois modalités employées dans la section 1.4. Dans la section 1.5, la fusion de ces trois percepts est détaillée selon un formalisme Bayésien. Dans la section 1.6, nous présentons les évaluations du détecteur multimodal. Enfin, dans la section 1.7, nous ouvrons la discussion quant à ce détecteur d'intentionnalité et d'éventuelles perspectives.

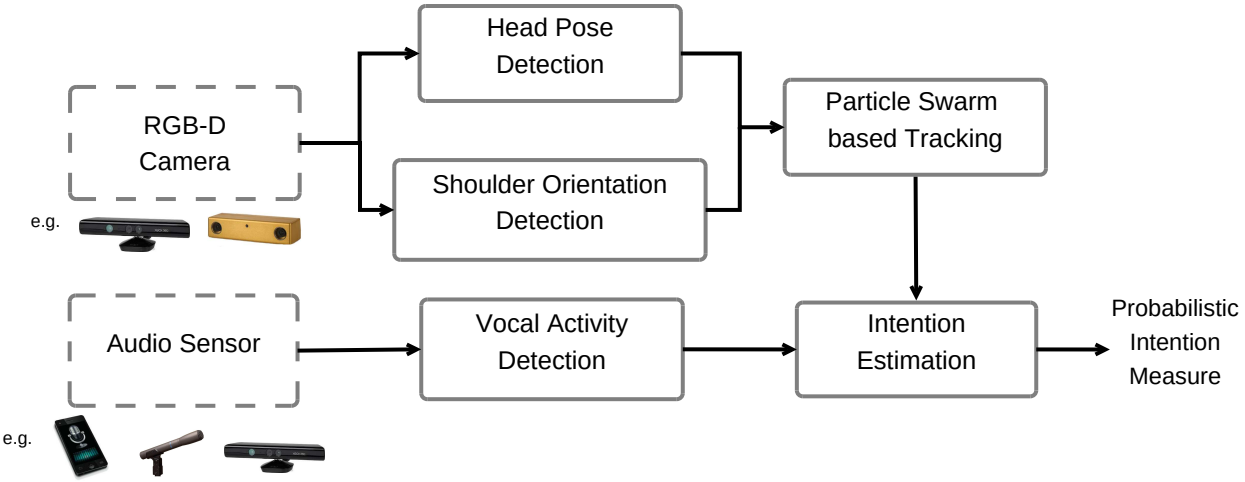


FIGURE 1.1 – Architecture complète du détecteur d'intentionnalité.

1.2 Etat de l'art

Comme décrit précédemment, nous définissons l'intentionnalité comme « l'intention de l'utilisateur d'initier une phase d'interaction proximale avec le robot ». Dans cette section, nous commençons donc par présenter les travaux relatifs à celle-ci dans la section 1.2.1. Dans la section 1.2.2, nous présentons un état de l'art sur les modalités visuelles employées dans notre détecteur.

1.2.1 Conceptualisation et perception de l'intentionnalité

Certains travaux liés à la perception de l'intentionnalité sont déterminants dans la façon dont nous avons défini celle-ci. Ainsi, Xiao et al. dans (Xiao et al., 2014) ont utilisé la reconnaissance de gestes pour en définir un certain nombre qui permettent d'initier ou de stopper une session d'interaction. D'autres gestes servent uniquement à ponctuer le langage ou à indiquer le déroulement d'une activité, telle que passer un coup de téléphone, ceux-ci ne participant pas de l'intentionnalité d'interaction. Les auteurs présentent certains gestes comme étant une information importante dans le déclenchement d'une session d'interaction.

Huber utilise quant à lui la position des pieds comme indicateurs de l'intentionnalité dans (Huber, 2013). Pour ce faire il se sert d'un capteur laser positionné au niveau du sol afin de percevoir l'orientation des pieds, ce qui permet entres autres de savoir si plusieurs personnes dans une pièce sont susceptibles d'interagir entre elles. Un algorithme SVM (Support Vector Machine) permet de classifier deux types de comportements avec une précision globale de 84%.

D'autres auteurs ont traduit l'intention comme le but sous-jacent à un comportement observé. Dans un contexte robotique, St. Clair et al. dans (Clair et al., 2010) utilisent l'orientation du visage et la position de l'utilisateur pour lever des ambiguïtés quant aux objets mentionnés durant l'interaction. Par exemple, lorsque plusieurs objets mentionnés par l'utilisateur sont identiques, l'orientation du visage est utilisée par le robot comme information déictique pour identifier le bon

objet et donc prendre une décision.

De la même manière, Tavakkoli et al. dans (Tavakkoli et al., 2007) utilisent un algorithme HMM qui permet de choisir entre trois comportements dans la rue : une rencontre entre deux passants, un passant en suivant un autre, ou bien deux passants qui se croisent sans interagir. Le HMM effectue une classification selon la position 3D des personnes suivies dans l'image. Un algorithme de soustraction de fond permet de faciliter le suivi. Ils obtiennent ainsi une précision globale de 84%.

Dans leurs travaux (Brochard et al., 2009), Brochard et al. utilisent l'orientation du regard comme information déictique en partant du principe que lorsqu'un objet est évoqué, l'utilisateur à tendance à le regarder simultanément. L'orientation du regard est extraite à l'aide de descripteurs SURF (Bay et al., 2008), qu'un algorithme de filtrage particulaire de type ICONDENSATION (Isard and Blake, 1998) va ensuite raffiner. Le fonctionnement optimal de cet algorithme suppose que la caméra soit positionnée face à l'utilisateur, ce qui pose problème dans notre contexte car l'intentionnalité doit être perçue à distance.

Rios-Martinez et al. dans (Rios-Martinez et al., 2012) présentent un système de planification semi-automatique de trajectoire. Celui-ci utilise la fusion entre l'orientation du visage de l'usager et le système de planification pour piloter une chaise roulante. L'orientation du visage est employée pour améliorer la détection d'obstacle mouvants et re-calculer la trajectoire en conséquence. Leur architecture est validée par différents scénarii impliquant la chaise roulante et des personnes marchant dans l'environnement.

Les auteurs Kuan et al. dans (Kuan et al., 2010) définissent l'intention comme la fusion entre un électromyogramme, la mesure d'angle du bras et la force exercée. Cette fois, l'intention est utilisée comme indicateur pour accompagner un patient équipé d'un exo-squelette dans son processus de rééducation. Le signal de l'électromiogramme est filtré pour en réduire le bruit. Un algorithme de classification vient ensuite décider si l'intention de l'utilisateur correspond à une flexion, une extension ou bien une excitation musculaire neutre.

Tous ces travaux définissent l'intention comme commande non-verbale d'un système robotique, soit comme information déictique. Toutefois, dans notre contexte, l'intentionnalité correspond au début d'une phase d'interaction permettant de rendre le robot pro-actif. La conceptualisation de l'intention dans les travaux pré-cités diverge donc de la nôtre.

Plus proche de notre application (Kulić and Croft, 2003) présentent un processus d'intention composé de deux paramètres : l'attention et l'approbation. Ce qu'ils définissent comme « attention » se rapproche de ce que nous appelons « intentionnalité ». Ils définissent cela comme une estimation de l'engagement d'une personne lors de l'interaction, permettant de savoir si la personne y participe, à la fois physiquement (orientation du corps, direction du regard) que de façon cognitive (expressions du visage, langage corporel). Les auteurs estiment l'attention à l'aide de signaux physiologiques, tels que la pression sanguine et le rythme cardiaque. Ils mesurent les résultats sur une échelle d'éveil (arousal) et de valence (Hudlicka and McNeese, 2002).

De manière similaire Ooko et al. dans (Ooko et al., 2011) se focalisent sur ce qu'ils appellent l'engagement de l'utilisateur. Ils détectent l'implication de l'utilisateur dans une conversation en cours à l'aide de l'orientation et la position du visage. Ils effectuent ensuite un seuillage de ces paramètres pour estimer l'engagement de l'utilisateur. Ils arrivent ainsi à 77% de classification

correcte.

Bascetta et al. quant à eux se focalisent sur l'intention d'un utilisateur en déplacement dans (Bascetta et al., 2011). Dans un contexte de coopération en robotique industrielle, ils essayent de déterminer si l'utilisateur va effectuer une tâche qui pourrait gêner les mouvements du robot et donc se mettre en danger. Un HMM permet de modéliser par apprentissage les zones de la pièce où l'utilisateur peut potentiellement interférer avec le robot, en observant le comportement de l'utilisateur dans des scénarii de coopération, de coexistence et d'interférence.

Étant donnés les travaux pré-cités, nous pouvons noter l'importance de l'orientation du corps, du visage et du regard. Comme nous voulons que notre détecteur puisse fonctionner au delà d'une distance de 2 mètres, nous nous sommes orientés vers une détection de l'orientation du visage par images de profondeur. Dans notre contexte, il n'est pas possible d'estimer l'orientation du regard car la plupart des algorithmes ne fonctionnent qu'avec le visage statique et proche de la caméra. Nous nous sommes ainsi axés sur l'orientation des épaules, cette information indiquant l'orientation générale du corps. Le capteur Kinect semble donc tout indiqué pour cela puisqu'il permet l'extraction du squelette de l'utilisateur à partir d'images de profondeur. La détection d'activité vocale vient en complément de ces détecteurs dans ce contexte de pré-interaction. Ceci est particulièrement novateur sachant qu'au moment de notre recherche, aucune étude ne l'utilisait pour la détection d'intentionnalité. Nous modélisons l'intentionnalité par un algorithme HMM souvent utilisé dans la littérature scientifique pour ce genre de tâches.

1.2.2 Estimation d'orientation du visage

Dans cette section nous nous intéressons aux différentes techniques qui permettent une estimation visuelle de l'orientation du visage qui renseigne notre détecteur d'intentionnalité. Nous nous intéressons tout d'abord aux techniques d'estimation 2D puis à l'estimation 3D.

Estimation à partir d'images 2D

Dans (Murphy-Chutorian and Trivedi, 2009), les auteurs ont étudié 37 techniques différentes de détection et estimation d'orientation du visage. Après un état de l'art exhaustif, ils recensent les techniques basées sur des modèles d'apparence actifs (AAM), des algorithmes basés sur la géométrie des visages, et des méthodes de détection par suivi visuel (tracking). Enfin, ils décrivent les méthodes de régression pour déterminer l'orientation du visage. Toutes leurs investigations se basent sur des travaux dans un espace colorimétrique RGB. Leur estimation se fonde donc à partir de données en 2D. Ils comparent ensuite les précisions affichées par chaque algorithme. Nous nous focalisons ici sur les méthodes par régression, celles-ci présentant le meilleur compromis entre vitesse de détection et précision. Elles font aussi partie des méthodes privilégiées depuis quelques années du fait de l'apparition des capteurs RGB-D à bas coût.

La régression requiert un apprentissage supervisé. Huang et al. utilisent une méthode de type SVM (Support Vector Machine) (Aizerman et al., 1964) dans leurs travaux (Huang et al., 1998), ce qui permet de classer chaque image de visage en 3 classes : une classe orientation gauche, une classe orientation de face, et une classe orientation droite avec des résultats proches de 100%.

Les réseaux de neurones ont aussi été très utilisés dans l'estimation d'orientation du visage par régression. (Seemann et al., 2004) Les niveaux de gris et les informations de profondeur extraites par un banc stéréoscopique ont été combinés à l'aide d'un réseau de neurones afin d'estimer l'orientation du visage avec moins de 20% d'erreurs dans 90% des cas. Un filtre de Kalman vient ensuite raffiner les résultats du détecteur seul.

De leur côté, Valenti et al. (Valenti et al., 2012) utilisent l'estimation de l'orientation du visage à l'aide d'un modèle cylindrique à partir d'une image RGB classique permettant de renforcer la détection des pupilles. L'estimation est initialisée par un détecteur de visage orienté de face. Un algorithme basé sur le flot optique et la re-projection par transformation affine permet de suivre la rotation du visage. La position des yeux est ensuite déduite de l'orientation du cylindre et comparée à la position renvoyée par un détecteur de pupilles. Cette technique permet à la fois de renforcer le détecteur de pupilles et l'estimation de l'orientation du visage. Ce dispositif souffre une erreur angulaire inférieure à 6 degrés. L'inconvénient majeur de cette technique est que l'utilisateur doit se mettre face à la caméra pour initialiser le dispositif.

Estimation à partir d'images 3D

Depuis l'apparition des capteurs RGB-D de nombreux travaux se mettent à exploiter l'information de profondeur. Ainsi, dans (Martin et al., 2014), les auteurs utilisent un détecteur de visage permettant de créer un modèle à partir d'un nuage de points. Ce nuage est ensuite suivi dans le temps, tout en raffinant l'estimation d'orientation grâce à un algorithme de type ICP (Iterative Closest Point, (Chen and Medioni, 1991)). Ils obtiennent ainsi une erreur angulaire inférieure à 6 degrés avec un taux de succès supérieur à 94%. L'inconvénient de cet algorithme est là encore l'obligation pour l'utilisateur de faire face à la caméra.

Toujours basés sur une estimation par modèles, les travaux des auteurs Padeleris et al. utilisent une image de profondeur de référence pour initialiser leur algorithme (Padeleris et al., 2012). Ils formalisent ensuite leur problème comme un processus d'optimisation continue en cherchant quelle pose du modèle correspond le plus à la nouvelle image. Ils obtiennent ainsi une précision inférieure à 3 degrés avec un taux de succès supérieur à 78%.

Fanelli et al. dans (Fanelli et al., 2013) utilisent des informations extraites d'un capteur Kinect pour estimer les trois angles et la position du visage dans l'espace. La régression est effectuée à l'aide d'un algorithme de Random Forest. Ils parviennent ainsi à une erreur angulaire moyenne inférieure à 6 degrés. La force de cet algorithme réside dans le fait qu'il peut directement détecter l'utilisateur sans initialisation particulière. Une technique très similaire est employée dans (Qiao and Dai, 2013).

Notre objectif étant de s'affranchir de la contrainte d'initialisation (position fixe devant la caméra), nous nous sommes orientés vers cette dernière approche. En effet, la précision obtenue grâce au capteur Kinect est amplement suffisante dans notre contexte applicatif (erreur inférieure à une dizaine de degrés). De plus, l'utilisation des images de profondeur permet d'obtenir la position relative de l'utilisateur par rapport au robot. Ce détecteur est aussi Open Source, ce qui constitue un autre avantage en permettant une implémentation rapide de la modalité d'orientation du visage. Enfin, cet algorithme fonctionne en temps réel, ce qui répond à notre besoin de réactivité pour une interaction homme-machine la plus naturelle possible.

1.3 Bibliothèques utilisées

Pour implémenter un détecteur d'intentionnalité qui réponde aux spécifications décrites précédemment, nous avons envisagé l'utilisation de plusieurs bibliothèques pour l'implémentation des modalités.

1.3.1 Gestion du capteur Kinect

Le détecteur d'intentionnalité est basé sur les informations issues d'un capteur Kinect, nous présentons ici les 3 bibliothèques les plus courantes qui permettent d'exploiter ce capteur :

Kinect SDK : c'est actuellement la bibliothèque officielle et donc la plus optimisée (Webb and Ashley, 2012). Cette bibliothèque créée par Microsoft est utilisable uniquement sous Windows. Elle permet une prise en charge complète du capteur Kinect tout en offrant la possibilité d'utiliser des algorithmes de génération de squelette et de détection d'orientation de visage.

OpenNI : c'est la deuxième bibliothèque la plus utilisée (PrimeSense, 2010). Celle-ci permet la prise en charge Open Source et multi-plateforme de la partie vidéo du capteur Kinect. Elle offre également quelques fonctionnalités supplémentaires, telles que la génération de squelette via le Middleware NITE. Cette bibliothèque a cependant été rachetée et n'est plus en développement.

OpenKinect : cette dernière est également Open Source et multi-plateforme (OpenKinect, 2010). Elle permet une gestion complète des fonctions de base du capteur, en offrant l'accès aux parties vidéo et audio, et au contrôle du moteur. Cependant, aucune fonctionnalité supplémentaire n'est présente pour le moment.

Nous nous sommes donc tournés vers la solution OpenNI, celle-ci offrant le meilleur compromis entre l'exploitation des fonctionnalités de base du capteur et la présence indispensable de la détection de squelette pour la modalité d'orientation des épaules. Les informations relatives à chaque bibliothèque sont récapitulées dans la table 1.1.

TABLE 1.1 – Table comparative entre les trois bibliothèques d'interface de capteurs RGB-D.

Bibliothèque	**Plateforme**	**Fonctionnalités**	**Capteurs**
Kinect SDK	Windows	images RGB et Profondeur, microphone, pilotage du moteur, détection de squelette	Kinect
OpenNI	Windows, Linux, Mac	images RGB et Profondeur, détection de squelette.	Kinect, X-tion
OpenKinect	Windows, Linux, Mac	images RGB et Profondeur, microphone, pilotage du moteur	Kinect

1.3.2 PocketSphinx

Dans cette section, nous détaillons la bibliothèque PocketSphinx et son intérêt pour notre problématique.

Décrite dans (Huggins-daines et al., 2006), celle-ci a été créée par le CMU. Elle est la seule bibliothèque de reconnaissance vocale OpenSource qui permette une liberté maximale sur le choix des grammaires et modèles acoustiques. Elle permet ainsi de créer ses propres modèles acoustiques et modèles de langages. De plus, elle est compatible avec les modèles du LIUM présentés dans (Deléglise et al., 2005) pour une utilisation en français. D'autres langues telles que l'anglais, l'espagnol ou l'allemand sont également disponibles.

Nous avons également utilisé le détecteur d'activité vocale de SphinxBase (compatible avec PocketSphinx) pour extraire en ligne les segments de parole et de non-parole d'un flux audio. Cela rend le détecteur compatible avec toute la chaîne de perception (pipeline) de l'utilisateur qui sera présentée dans le chapitre 3.

1.4 Perception multimodale de l'utilisateur

Cette section décrit les trois modalités utilisées par le détecteur d'intentionnalité (voir la figure 1.2). Á savoir, la détection d'orientation du visage, des épaules et enfin, d'activité vocale.

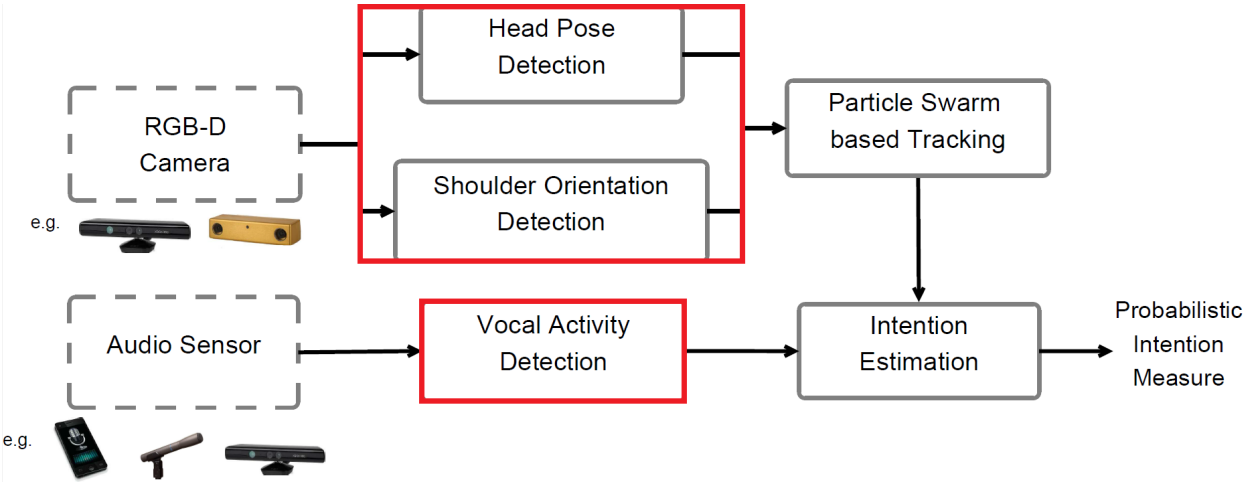


FIGURE 1.2 – Architecture complète du détecteur d'intentionnalité. Le rectangle rouge met en avant les trois modalités utilisées en entrée du détecteur.

1.4.1 Modalité 1 : orientation du visage

La première modalité que nous avons choisie est visuelle : elle repose sur un algorithme de régression permettant de détecter et d'estimer l'orientation et la position du visage dans une image

de profondeur. Nous estimons ainsi un vecteur de six dimensions : trois dimensions angulaires et trois dimensions spatiales. Pour effectuer cette estimation nous avons privilégié l'approche présentée dans (Fanelli et al., 2011). Ce module de régression est basé sur un algorithme de type Random Forest. Cet algorithme utilise des images de profondeur fournies par un capteur RGB-D (Kinect, X-Tion, etc.). L'orientation est donc estimée par rapport à la caméra profondeur. L'ensemble des paramètres utilisés par l'algorithme et évoqués par la suite sont résumés dans la table 1.2.

TABLE 1.2 – Paramètres utilisés par l'algorithme de Random Forest en apprentissage et détection

Paramètre	Description
Entrainement	
d_{RF}	Profondeur de l'arbre. L'arbre créé aura au maximum $(2d_{RF}+1)$ feuilles.
N_{arbres}	Nombre d'arbres dans la forêt.
N_{ROI}	Nombre de région d'intérêt ou imagettes par image.
N_{images}	Nombre d'images dans le dataset.
$N_{dataset}$	Nombre total d'imagettes.
$\epsilon_{desc}^{(i)}$	Sous-ensemble de descripteurs sélectionné dans un ensemble de N_{desc} pour le nœud (i).
$\epsilon_{seuils}^{(i)}$	Sous ensemble de seuils sélectionné dans un ensemble de N_{seuils} pour le nœud (i).
Régression	
S_{det}	Seuil sur les ensembles de points (clusters) pour générer une détection.
d_{max}	Distance maximum pour la détection.

Apprentissage des arbres

L'algorithme de Random Forest consiste à créer une « forêt » d'arbres de décisions par apprentissage automatique. Ce type d'algorithme à l'avantage d'être extrêmement rapide, aussi bien en terme d'apprentissage, que de détection. Ce dernier critère est important : l'interaction homme-machine doit se rapprocher du temps réel et réduire le temps de latence pour l'utilisateur. Dans cette section, nous détaillons la phase d'apprentissage.

L'apprentissage automatique supervisé repose sur l'exploitation d'un corpus de données annotées. Dans l'application de (Fanelli et al., 2013), le *Biwi Kinect Head Pose Dataset* a servi de corpus. Dans chaque image de profondeur de ce corpus, le premier plan a été segmenté afin de ne garder que les pixels contenant le visage des personnes. Des régions d'intérêt ou imagettes ont été extraites des visages et associées aux orientations annotées. Un nouveau corpus a été ainsi conçu, et se caractérise comme suit : $N_{dataset} = N_{images} * N_{ROI}$ (N_{ROI} correspond au nombre de régions d'intérêts par images, et N_{images} correspond au nombre d'image). L'image intégrale a été calculée

pour chaque région d'intérêt, et la valeur de chaque pixel $ii(x, y)$ aux coordonnées (x, y) est extraite grâce à l'équation (1.1).

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \tag{1.1}$$

Comme le nom Random Forest l'indique, la forêt est composée d'arbres construits de manière aléatoire par la sélection d'un sous-ensemble d'imagettes.

Dans cet algorithme, un arbre est construit comme un arbre de décision. La racine (base) de l'arbre reçoit une imagette ou région d'intérêt. Dans chaque nœud, un test binaire est effectué et le subdivise en deux branches. Chaque branche mène à un nouveau nœud et ainsi de suite. Lorsqu'un critère d'arrêt est satisfait, l'arbre crée une feuille pour terminer la branche au lieu de créer un nouveau nœud. La profondeur de l'arbre est un paramètre libre de l'algorithme qui fait office de critère d'arrêt, noté ici d_{RF}. Dans cette application, d_{RF} est fixé à 10 pour éviter le sur-apprentissage tout en conservant une précision correcte. La figure 1.3 illustre un arbre de profondeur 3.

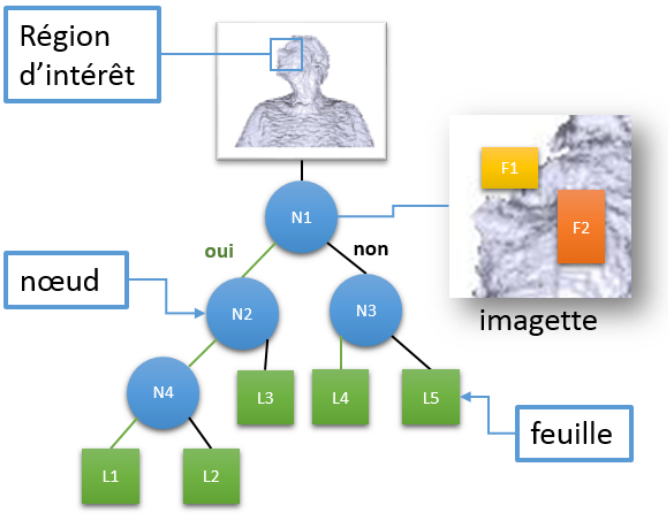


FIGURE 1.3 – Représentation d'un arbre de profondeur $d_{RF} = 3$. Les ronds bleus représentent les nœuds. Les carrés verts représentent les feuilles. L'imagette représente l'extraction des descripteurs (différence d'intensité entre le rectangle F1 et F2) lors du passage dans le premier nœud.

L'apprentissage est donc effectué au niveau de chaque nœud. Lorsque la base de données composée des $N_{dataset}$ imagettes arrive au niveau d'un nœud, un sous-ensemble ϵ_{desc} de N_{desc} descripteurs est sélectionné. Dans cette application, les descripteurs correspondent à une différence de profondeur entre deux régions d'intérêt au sein de l'imagette. Ces variations locales de profondeur permettent de repérer des zones du visages présentant des contrastes en profondeur (arêtes du visage, yeux, etc.). Ces descripteurs peuvent être assimilés à des descripteurs de HAAR généralisés comme l'ont présenté Viola et Jones (Viola and Jones, 2001). Un ensemble de seuils ϵ_{seuils} est

ensuite sélectionné aléatoirement de manière à classer l'ensemble des imagettes par rapport aux descripteurs choisis. Le choix du couple descripteur/seuil pour le nœud (i) s'effectue de manière à minimiser les fonctions de coûts (1.2) et (1.3). Une fois ce couple déterminé, le corpus est divisé en deux sous-catégories via cette règle de décision binaire, et l'apprentissage se poursuit le long des deux nouvelles branches créées.

$$IG = H(\mathcal{P}) - (\omega_L H(\mathcal{P}_\mathcal{L}) + \omega_R H(\mathcal{P}_\mathcal{R})) \tag{1.2}$$

La fonction de coût (1.2) permet de maximiser le gain d'information (IG) défini comme l'entropie différentielle entre l'ensemble de patchs d'un nœud parent $\mathcal{H}(\mathcal{P})$ d'un côté et l'entropie pondérée calculée dans les patchs des nœuds fils $\mathcal{H}(\mathcal{P}_L)$ (branche gauche) et $\mathcal{H}(\mathcal{P}_R)$ (branche droite) de l'autre, là où $\omega_{i\in L,R} = \frac{|\mathcal{P}_i|}{|\mathcal{P}|}$. Cette fonction favorise les descripteurs permettant une séparation franche entre les patchs représentant une image positive (c'est-à-dire une image représentant un visage) et une image négative (ne présentant aucun visage).

$$IG = log(|\Sigma^v| + |\Sigma^a|) - \sum_{i\in\{L,R\}} \omega_i log(\Sigma_i^v + \Sigma_i^a) \tag{1.3}$$

La fonction de coût (1.3) permet de minimiser la matrice de covariance des vecteurs d'état dans les nœuds fils. Σ^v représente la matrice de la partie spatiale de l'estimation et Σ^a représente la partie relative à l'orientation du visage. Cela permet non seulement de séparer les différentes orientations, mais aussi d'augmenter la précision des estimations en diminuant la covariance des feuilles. Une feuille est créée lorsque la branche atteint la profondeur maximale d_{RF} ou bien lorsque le nouveau nœud ne contient pas assez d'imagettes pour permettre une classification efficace. Un arbre est ainsi terminé et il contient $d_{RF} * 2 + 1$ feuilles au maximum, dans l'hypothèse où toutes les branches atteignent la profondeur maximale. En itérant ce processus sur plusieurs arbres on obtient ainsi une forêt qui permet d'estimer les orientations du visage avec précision. Dans cette application, le nombre d'arbres est fixé à $N_{arbres} = 10$.

Une fois la forêt d'arbres de décisions apprise par l'algorithme, celle-ci peut enfin être utilisée pour la régression. Chaque feuille donnera ainsi une orientation, la détection du visage se faisant à partir d'un ensemble de feuilles à l'aide d'un algorithme de type « camshift » (Cheng, 1995).

Détection d'orientation

La phase de détection est réalisée de la même manière que l'apprentissage, les seuils étant fixés pour chaque nœud lors de cette dernière phase. La régression est effectuée pour chaque image indépendamment de la détection précédente.

Lorsque l'image est reçue, elle est découpée en imagettes dont chacune est envoyée à la forêt où elle sont associées à une feuille par arbre. Nous obtenons alors dix résultats de régression pour chaque imagette. L'ensemble des positions et orientations possibles sont associées à une confiance résultant des covariances Σ^v et Σ^a dans chaque feuille.

Cet ensemble est ensuite utilisé par un algorithme de type « camshift » qui génère plusieurs sous-ensembles, chacun donnant une position et une orientation du visage. L'algorithme de régression

devient ainsi un détecteur grâce à un seuillage sur la taille de ces sous-ensembles. Pour le résultat, nous ne gardons que le sous-ensemble le plus important. Si celui-ci est inférieur à un seuil S_{det}, aucune détection n'est perçue pour l'image de profondeur considérée. Un seuillage est aussi appliqué sur la profondeur avec le paramètre d_{max}. Aussi, tous les pixels de profondeur supérieurs à ce seuil seront ignorés lors de la détection.

1.4.2 Modalité 2 : orientation des épaules

Comme expliqué dans la section 1.3.1, la modalité d'orientation des épaules est assurée par le module de génération de squelette de la bibliothèque OpenNI (PrimeSense, 2010). Comme pour la modalité précédente, la génération du squelette de l'utilisateur est effectuée à l'aide d'un algorithme de régression de type Random Forest. Au lieu d'estimer l'orientation et la position d'un visage, cet algorithme va estimer la position 3D des différentes parties du corps, notamment des épaules.

Le squelette ainsi généré comporte quinze articulations. Le calcul décrit par l'équation (1.4) a été appliqué sur les deux articulations représentant les épaules pour obtenir l'orientation azimutale relative au capteur RGB-D. X_R et Z_R représentent les coordonnées de l'épaule droite, X_L et Z_L sont les coordonnées de l'épaule gauche, et θ contient l'angle estimé.

$$\begin{aligned} NORM &= \sqrt{(X_R - X_L)^2 + (Z_R - Z_L)^2} \\ \theta &= acos(\frac{X_R - X_L}{NORM}) \\ \theta &= \begin{cases} \theta & \text{si } acos(\frac{Z_R - Z_L}{NORM}) > \frac{\pi}{2} \\ -\theta & \text{si } acos(\frac{Z_R - Z_L}{NORM}) \leq \frac{\pi}{2} \end{cases} \end{aligned} \tag{1.4}$$

1.4.3 Modalité 3 : détection d'activité vocale

La modalité de détection d'activité vocale permet de détecter la présence de parole dans un flux audio. Un tel algorithme est normalement utilisé pour segmenter un signal audio en zones de parole et de non-parole qui déclenche ensuite un algorithme de transcription sur les zones de parole. Contrairement à ce dernier, l'algorithme de détection d'activité vocale n'est pas capable de reconnaître les mots prononcés mais peut fonctionner à plus grande distance du microphone.

$$E_S = \sum_{t=1}^{N} |x(t)|^2 \tag{1.5}$$

Dans cette modalité, nous avons utilisé l'algorithme présent dans la bibliothèque PocketSphinx. Celui-ci est construit à partir d'un seuillage de l'énergie du signal, défini par l'équation (1.5). $x(t)$ représente un échantillon du signal à l'instant t, N est la taille de la fenêtre dans laquelle est calculée l'énergie.

Avant toute détection, l'utilisateur doit rester silencieux afin de permettre le calibrage de l'algorithme : celui-ci doit prendre en compte le fond sonore. L'énergie est calculée à partir de fenêtres

glissantes de 512 échantillons avec un recouvrement de 256 échantillons. Le signal est échantillonné à 16 kHz et quantifié sur 16 bits.

1.5 Fusion audio-visuelle : formalisation

Rappelons que la finalité de ce chapitre est de fusionner divers percepts visuels et sonores relatifs à la détection d'intentionnalité. Cette section présente la fusion des trois percepts décrits précédemment. Conformément à notre architecture, cette section détaille la partie encadrée sur la figure 1.4.

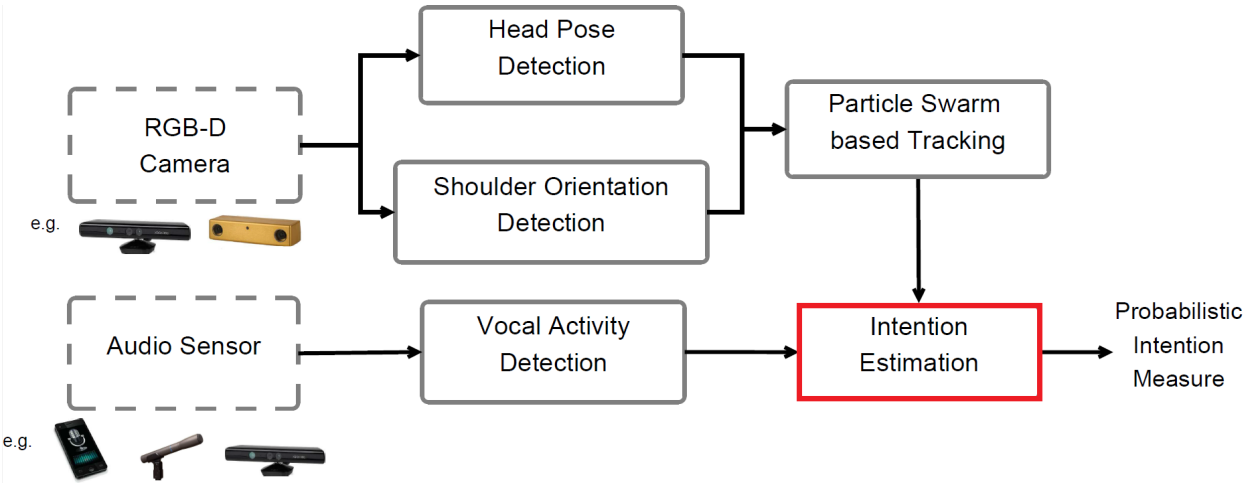


FIGURE 1.4 – Architecture complète du détecteur d'intentionnalité. En rouge, la partie « décision » du détecteur détaillée dans cette section.

Nous avons assimilé la détection d'intentionnalité au détecteur qui permet de déceler l'intention de communication d'un utilisateur. Afin de mettre au point ce détecteur, nous nous sommes appuyés sur plusieurs principes :

1. Lorsqu'une personne veut communiquer avec quelqu'un d'autre, elle initie rarement l'interaction en lui tournant le dos. *A priori*, une intentionnalité est donc plus probable lorsque l'utilisateur est orienté vers le robot ou directement face à lui.
2. Lorsque l'utilisateur veut commencer à s'adresser au robot, il ne le nomme pas nécessairement, en particulier si l'utilisateur est proche et orienté dans sa direction (regard, visage, et/ou épaules).

Suivant ces principes, la probabilité de détecter une intentionnalité est maximale lorsque l'utilisateur est orienté vers le robot (orientation des épaules et du visage)et s'adresse à lui. A contrario, le robot ne doit pas interpréter comme une intentionnalité un simple regard.

Il y a donc une nécessité de recouper plusieurs informations (orientations et parole). Par exemple, dans le cas d'une émission de radio, le détecteur d'activité vocale va détecter des zones de parole, mais en l'absence d'orientation de l'utilisateur vers le robot, le détecteur ne se déclenchera pas.

Pour décrire cette triple indication, nous avons utilisé un formalisme bayésien à l'aide d'un modèle de Markov caché (Hidden Markov Model : HMM). Nous avons donc modélisé l'intentionnalité comme un processus Markovien à états discrets indiquant la présence ou non d'une intentionnalité. Ces états étant cachés, ils sont observés de manière indirecte grâce à l'orientation du visage, des épaules et à la détection d'activité vocale. Ces observations sont hybrides : l'orientation du visage et des épaules sont des processus à états continus et leurs observations sont modélisées par une Gaussienne multivariée, tandis que la détection d'activité vocale est un processus à états discrets. La distribution d'observation de l'orientation est maximale lorsque la tête et les épaules se positionnent en direction du robot. Elle a pour moyenne le vecteur nul par rapport au capteur RGB-D. Étant donné que l'orientation est un processus bruité, une étape de filtrage supplémentaire s'impose et permet de rendre le processus plus robuste. Cette composante sera détaillée dans le chapitre suivant. L'observation de la détection d'activité vocale est modélisée par une observation : 1 ou une absence d'observation : 0

$$P\left(\boldsymbol{x_t}|\boldsymbol{Z_{1:t}}\right) = \eta P\left(\boldsymbol{z_t^1}|\boldsymbol{x_t}\right) P\left(\boldsymbol{z_t^2}|\boldsymbol{x_t}\right) \sum_{\boldsymbol{x_{t-1}=1}}^{N} P(\boldsymbol{x_t}|\boldsymbol{x_{t-1}}) P(\boldsymbol{x_{t-1}}|\boldsymbol{Z_{1:t-1}}) \tag{1.6}$$

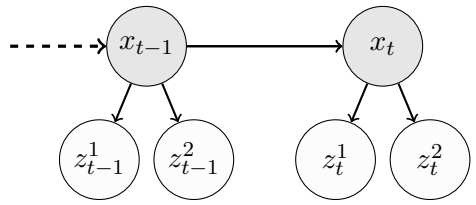


FIGURE 1.5 – Modèle graphique probabiliste utilisé pour l'estimation d'intentionnalité.

Nous obtenons alors le modèle graphique décrit dans la figure 1.5, où $\boldsymbol{x_{t-1}}$ et $\boldsymbol{x_t}$ représentent les états discrets de l'intentionnalité aux instants $t-1$ et t. $\boldsymbol{z_{t-1}^1}$ et $\boldsymbol{z_t^1}$ sont les densités observées à partir des vecteurs d'états estimés à l'aide du filtrage par essaim de particules (présenté dans le chapitre 2). Enfin, $\boldsymbol{z_{t-1}^2}$ et $\boldsymbol{z_t^2}$ sont les observations binaires du détecteur d'activité vocale. En supposant que les observations sont dépendantes de l'état précédent et en utilisant la formule de Bayes conditionnellement à toutes les observations jusqu'à l'instant t, $\boldsymbol{Z_{1:t}}$, la distribution *a posteriori* sur l'état $P(\boldsymbol{x_t}|\boldsymbol{Z_{1:t}})$ s'exprime selon l'équation (1.6) où $P(\boldsymbol{x_t}|\boldsymbol{x_{t-1}})$ représente la distribution de transition, et η est un facteur de normalisation.

Le HMM n'ayant que deux états dans notre application, la distribution de transition est représentée par une matrice carrée de dimension 4. La matrice d'observation du détecteur d'activité vocale est aussi modélisée par une matrice carrée de même dimension car ce détecteur ne peut se

trouver que dans deux états. Enfin, les observations de l'orientation du visage et des épaules étant à états continus, elles sont représentées par un modèle Gaussien multivarié de dimension 4 (trois angles pour le visage et un angle pour les épaules).

1.6 Fusion audio-visuelle : évaluation et discussion

Cette section présente les expériences et résultats associés au détecteur décrit précédemment.

Pour évaluer les performances de ce détecteur, trois bases de données ont été réalisées. Deux de ces bases (notées I et III dans la suite) ont été acquises en utilisant un robot PR2 (figure 1.6b) en situation d'interaction dans l'appartement ADREAM du LAAS-CNRS (figure 1.6a), simulant ainsi un environnement humain. Ces deux bases de données sont essentielles pour évaluer correctement notre détecteur qui pourra alors initier une session d'interaction proximale. La troisième base de données (notée II dans la suite) a été acquise en utilisant un capteur Kinect ainsi qu'un Smartphone dans un bureau du LAAS-CNRS. Cette différence d'environnement visuel et sonore permet de généraliser l'application de notre algorithme à d'autres situations d'interaction. Les tailles de ces bases de données varient de 2700 images à 3500 images (acquises à 30 ips). Elles sont constituées d'images RGB-D et de flux audio mono 16 bits échantillonnés à 16 kHz. Lors de ces sessions, les utilisateurs sont assis à une distance d'environ trois mètres du capteur RGB-D et à un mètre du smartphone. Ils montrent leur intention de communication en se tournant vers le robot et/ou en lui parlant. Par exemple, « J'aimerais que tu m'aides ». Les bases de données ainsi créées ont été annotées manuellement pour déterminer les plages d'intentionnalité que le robot doit effectivement détecter.

(a) appartement ADREAM (b) PR2

FIGURE 1.6 – Conditions d'acquisition en environnement humain au LAAS-CNRS.

Notre détecteur a été évalué en termes de pourcentages de vrais positifs(True Positive Rate : TPR), mais aussi de pourcentage de fausses alarmes (False Alarm Rate : FAR), de détection moyenne prématurée (Average Early Detection : AED) et de durée moyenne correcte (Average Correct Duration : ACD).

- Le TPR représente le rapport entre le nombre de détections correctes sur le nombre total d'images où l'intentionnalité est labellisée comme active.
- Le FAR est le rapport entre le nombre de détections labellisées comme positives par notre détecteur alors que la vérité terrain est labellisée comme négative sur le nombre total d'images labellisées négatives.
- La mesure AED est le temps moyen de décalage entre le début d'une plage positive dans la vérité terrain et le début d'une détection.
- L'ACD est le rapport entre le temps total d'une détection sur le temps total de détection de la vérité terrain pour cette session d'intentionnalité. La moyenne des différentes plages est ensuite calculée.

Les résultats présentés dans le tableau 1.3 montrent l'évaluation de ces différentes mesures sur les bases de données I et II. La base de données III acquise avec le PR2 a été utilisée pour l'apprentissage du HMM. Nous obtenons ainsi les paramètres dans l'équation (1.7) pour la matrice d'observation liée à la détection d'activité vocale. L'équation (1.8) décrit les observations liées à l'orientation du visage et des épaules, où Σ est diagonale de valeur 100 et de dimension 4, car ce sont des processus à états continus. Et enfin l'équation (1.9) représente la matrice de transition, le HMM étant par définition un processus à état discret.

$$P\left(z_t^2|x_t\right)=\begin{bmatrix}0.30 & 0.75\\ 0.70 & 0.25\end{bmatrix} \tag{1.7}$$

$$P(z_t^1|x_t)=\mathcal{N}\left(z_t^1;0,\Sigma\right) \tag{1.8}$$

$$P(x_t|x_{t-1})=\begin{bmatrix}0.990 & 0.017\\ 0.010 & 0.983\end{bmatrix} \tag{1.9}$$

Les résultats du tableau 1.3 sont affichés pour la détection d'activité vocale seule (VAD), l'orientation du visage et des épaules seule (RGB-D), et la fusion des trois modalités par HMM. Un exemple de comportement du détecteur est affiché figure 1.7.

Nous pouvons ainsi remarquer que la fusion des différentes modalités donne de meilleurs résultats dans un contexte multimodal pour toutes les mesures sauf pour le score AED. Ceci peut être expliqué par le fait que le détecteur d'activité vocale a tendance à se déclencher trop tard ou encore par le fait que l'utilisateur ne s'est pas orienté assez tôt vers le robot. Cependant, ce retard ne pose en pratique pas de problèmes, car il ne retarde le déclenchement du détecteur que d'une à deux secondes. Nous obtenons ainsi entre 72% et 80% de vrais positifs pour la fusion multimodale avec un taux de fausses alarmes situé entre 14% et 9%, alors qu'une modalité seule ne se situe qu'entre 48% et 72% de vrais positifs et entre 12% et 66% de fausses alarmes. Sur la base de données acquise avec le robot (base I), la détection est effectuée avec une latence de 20% en moyenne, et détecte l'intentionnalité pendant

74% en moyenne des plages labellisées comme détection dans la vérité terrain. La figure 1.7 illustre le fonctionnement de notre détecteur sur un sous-ensemble de la base de donnée.

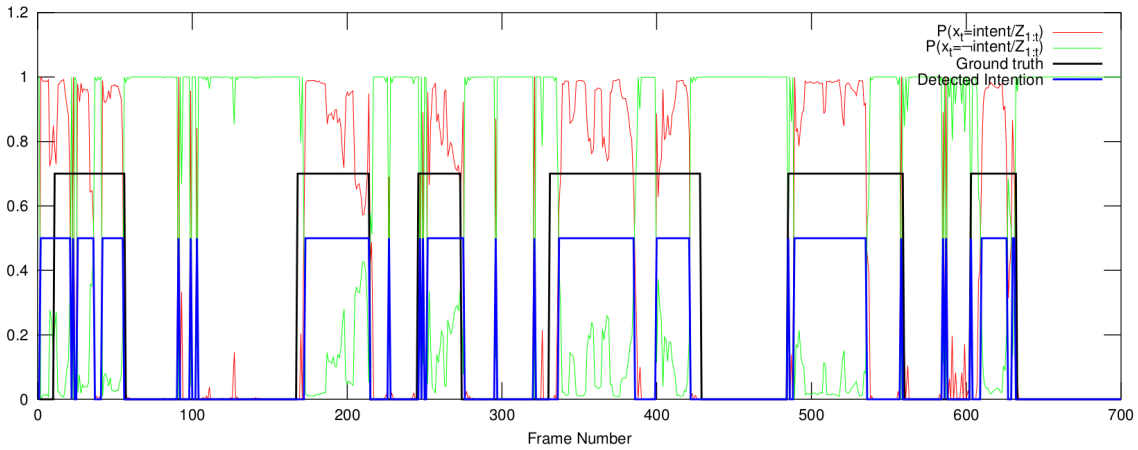


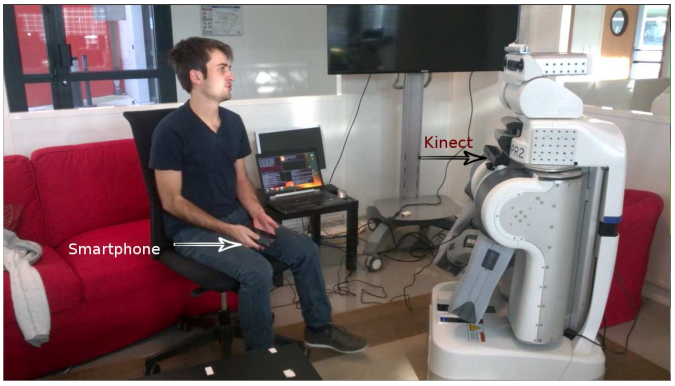
FIGURE 1.7 – La courbes bleue noire représentent respectivement la sortie de notre détecteur et la vérité terrain. Les courbes rouge et verte décrivent respectivement la probabilité de détecter ou non-détecter une intentionnalité, montrant l'évolution des deux états du HMM au cours du temps.

	TPR		FAR	
	I	II	I	II
VAD	0.48 (0.02)	0.56 (0.01)	0.66 (0.04)	0.50 (0.01)
RGB-D	0.68 (0.05)	0.72 (0.03)	0.26 (0.03)	0.12 (0.03)
Multimodal	**0.72** (0.03)	**0.80** (0.02)	**0.14** (0.04)	**0.09** (0.04)
	AED		ACD	
	I	II	I	II
VAD	**0.01** (0.00)	**0.03** (0.06)	0.33 (0.02)	0.56 (0.02)
RGB-D	0.26 (0.06)	0.10 (0.04)	0.64 (0.12)	0.73 (0.02)
Multimodal	0.20 (0.08)	0.10 (0.04)	**0.74** (0.06)	**0.77** (0.03)

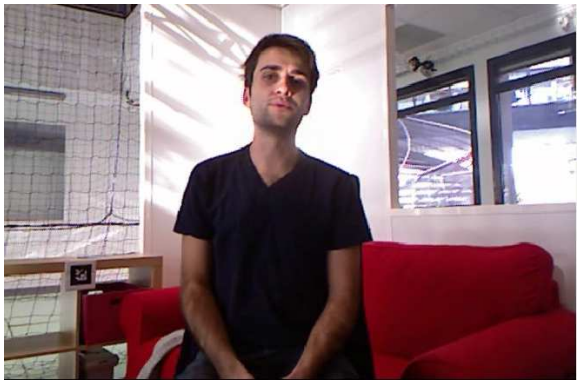
TABLE 1.3 – Résultats de la détection d'intentionnalité sur les bases de données I et II, décrits sous la forme « moyenne(écart-type) », basés sur une moyenne de dix analyses.

La figure 1.8a représente un extrait de la base de données I acquise dans un contexte d'interaction avec le robot. La figure 1.8b montre l'image couleur perçue par le capteur RGB-D embarqué sur le robot. La figure 1.8c représente le nuage de points extrait de l'image de profondeur où sont superposées les orientations du visage et des épaules de l'utilisateur. Enfin, la figure 1.8d montre la sortie de notre système. Les courbes bleue et noire représentent respectivement l'activité vocale

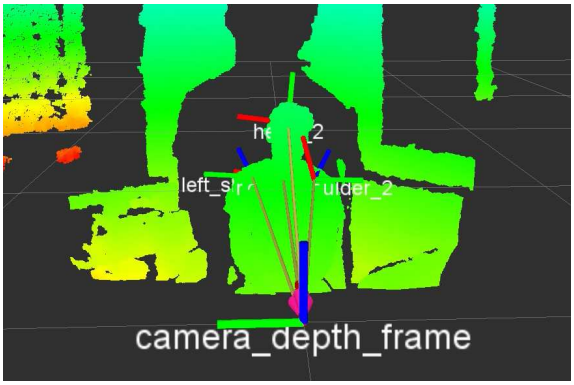
et la probabilité d'intention. La courbe bleue représente l'évolution de sortie du détecteur d'intentionnalité. L'axe des abscisses représente le temps affiché en nombre d'images (capteur à 30 ips). L'unité de l'axe des ordonnées est une probabilité $\in [0, 1]$ ou une détection $\in [\![0, 1]\!]$.



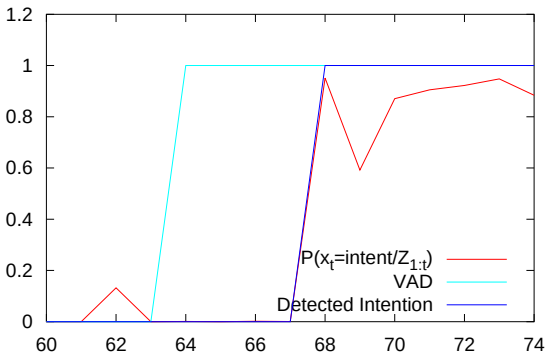
(a) Situation homme-robot



(b) Vue RBD du capteur Kinect



(c) Nuage de points extraits du capteur Kinect



(d) Affichage de la mesure d'intentionnalité

FIGURE 1.8 – Illustration du fonctionnement du détecteur d'intentionnalité (extrait de la base de donnée I).

1.7 Conclusion

L'intentionnalité, définie comme l'intention de l'utilisateur de communiquer avec un robot, est une notion essentielle dans un domaine tel que la robotique d'assistance à domicile non intrusive. Nous avons donc construit ce détecteur à partir de modalités visuelles, *i.e.* l'estimation d'orientation du visage et l'estimation d'orientation des épaules, ainsi que sur une modalité acoustique, *i.e.* la détection d'activité vocale. Assortie d'un algorithme de filtrage par essaim de particules (PSOT, voir le chapitre 2) et d'un algorithme de type Modèle de Markov Caché (HMM) pour modéliser le détecteur, l'architecture que nous avons obtenue permet de percevoir cette notion abstraite.

Nous avons acquis trois bases de données annotées dans deux contextes audios différents (bureau et salle d'expérience robotique), ce qui nous a permis de constater que notre détecteur d'intention-

nalité donne des résultats robustes avec un fort taux de vrais positifs (>72%) et un faible taux de fausses alarmes (<14%).

De nombreuses améliorations laissent la porte ouverte à des investigations complémentaires. Tout d'abord, de façon intuitive, le remplacement de chaque modalité (visuelle ou acoustique) par une modalité équivalente et plus performante devrait entraîner une amélioration tangible du détecteur. Nous pourrions aussi envisager de dupliquer le traitement de chaque modalité par des algorithmes utilisant différentes techniques. Par exemple, plusieurs détecteurs d'activité vocale ayant différentes sensibilités ou étant basés sur d'autres descripteurs audios pourraient être spécifiques à différentes plages de détection (longue distance, courte distance, environnement bruité, etc.). De plus nous nous sommes limités à trois modalités perceptuelles. Cela est principalement dû au fait que nous avons privilégié les capteurs embarqués sur le robot dans le cadre du projet RIDDLE. Nous pourrions envisager d'améliorer les performances de notre détecteur en y ajoutant par exemple un algorithme de séparation de sources permettant de filtrer les bruits environnants. Une telle solution a été présentée dans (Simon and Vincent, 2015).

Les contributions de ce chapitre ont permis une publication dans la conférence IEEE-ICME (Mollaret et al., 2015).

Chapitre 2

Suivi visuel du haut du corps

2.1 Introduction

Le suivi de cible est une thématique étudiée en vision par ordinateur de manière récurrente. Dans la plupart des cas, le suivi visuel permet d'ajouter une cohérence spatio-temporelle à des détections éparses, tout en filtrant les données incohérentes. Dans le cas du suivi multi-cibles, une étape de suivi visuel permet de lever les ambiguïtés d'identification lorsque plusieurs cibles sont détectées au même instant. Il permet aussi d'écarter l'une des hypothèses dans le cas de multiples détections, dans un contexte mono-cible.

Dans le chapitre précédent, nous avons présenté notre détecteur d'intentionnalité. Celui-ci se fonde sur trois modalités, deux visuelles et une sonore. Les modalités visuelles reposant sur des détections une image après l'autre, nous avons ajouté un algorithme de filtrage par essaim de particules (Particle Swarm Optimization : PSO dans la suite) pour créer une cohérence temporelle entre les détections tout en gagnant en précision. Ce chapitre se tourne vers la problématique de filtrage en vision par ordinateur.

Pour rappel, les détections sont fournies par l'algorithme de détection d'orientation du visage présenté dans (Fanelli et al., 2011) et l'estimation d'orientation des épaules à l'aide de l'algorithme de *skeleton fitting* de la bibliothèque OpenNI (PrimeSense, 2010). Nous avons donc un vecteur d'état de dimension 7 à filtrer. Par rapport à notre architecture de détecteur d'intentionnalité, nous nous situons dans la partie « filtrage » encadrée en rouge sur la figure 2.1.

Dans ce chapitre nous faisons aussi état de notre contribution en matière de suivi visuel. L'algorithme de filtrage que nous avons développé, inspiré du PSO, est encore peu exploré dans la communauté Vision en ce qui concerne les problèmes de suivi de cible. Il a permis d'obtenir des performances intéressantes dans notre contexte applicatif.

Dans la section 2.2, nous présentons un état de l'art des techniques d'optimisation et de filtrage qui sont en lien avec notre problématique. Dans la section 2.3, nous détaillons le formalisme associé aux algorithmes approchant notre méthode de filtrage. Puis dans la section 2.4, nous décrivons les expériences menées afin de valider notre approche de suivi visuel et les résultats associés. Enfin la section 2.5 conclut ce chapitre sur filtrage en vision par ordinateur.

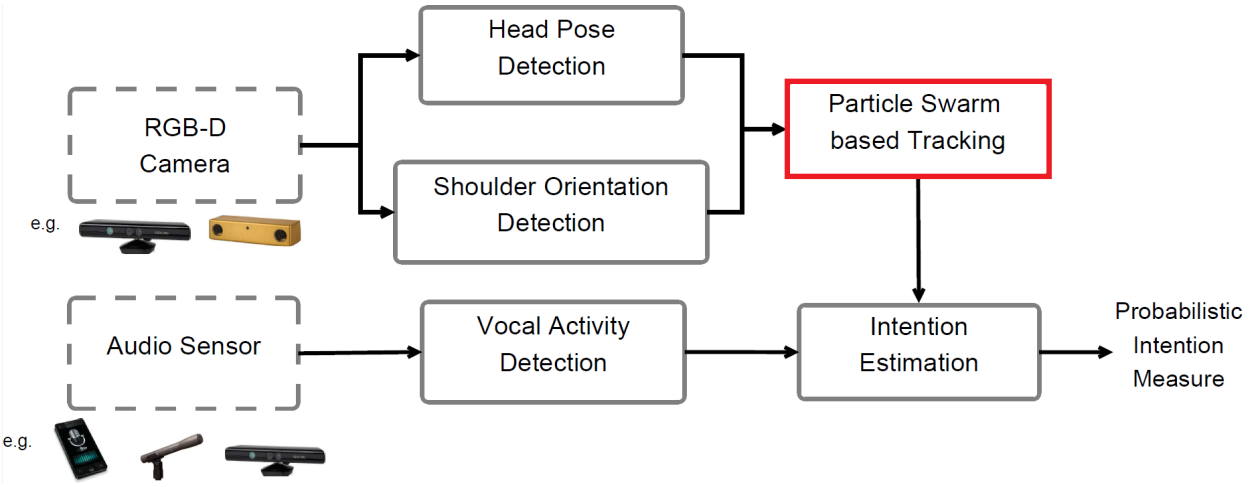


FIGURE 2.1 – Architecture complète du détecteur d'intentionnalité. Le cadre rouge désigne la partie « filtrage » détaillée dans ce chapitre.

2.2 État de l'art

L'état de l'art ayant contribué à notre algorithme de suivi visuel, nous avons choisi de décrire dans la section 2.2.1 quelques algorithmes communément employés dans la communauté de suivi visuel. La section 2.2.2 présente ensuite quelques algorithmes d'optimisation pour faire le lien avec la section 2.2.3 qui se focalise sur les techniques hybrides alliant les atouts de l'optimisation et du filtrage.

2.2.1 Suivi visuel

En vision par ordinateur le filtrage est souvent une étape indispensable pour pouvoir gagner une cohérence temporelle et supprimer les détections incohérentes produites par les différents capteurs. Il peut s'agir des informations de distance renvoyées par un capteur à ultrasons, ou bien encore des détections d'un algorithme de vision par ordinateur. Dans notre contexte applicatif, nous filtrons au sein d'un même processus les percepts extraits des estimations d'orientation des épaules et de visage.

En suivi visuel, un des algorithmes les plus connus et les plus employés dans la littérature scientifique est le filtre de Kalman. Une étude de Chen et al. dans (Chen, 2012) résume et compare différentes implémentations de cet algorithme. Le filtre de Kalman originel permet de filtrer des processus linéaires et gaussiens. Il est indiqué dans une grande variété d'applications telles que la vidéo surveillance et le positionnement par satellite. Cependant, dès que le modèle de la dynamique du système se complexifie et que des non-linéarités apparaissent dans la modélisation, il faut recourir à d'autres implémentations comme le filtre de Kalman sans parfum (Unscented Kalman Filter : UKF) présenté dans (Wan and Merwe, 2000) qui théorise alors l'utilisation du filtre de Kalman

pour des modèles non-linéaires et des processus non-gaussiens.

Un autre formalisme couramment utilisé est le filtre particulaire introduit par Doucet et al. dans (Doucet and Gordon, 1999). Aussi connu sous le nom de méthode séquentielle de Monte Carlo, celui-ci dérive des Méthodes Combinatoires de Monte Carlo (MCMC). L'avantage de ce type de filtre est qu'il est adapté aux dynamiques *a priori* non-linéaires ainsi que des distributions *a posteriori* multimodales, contrairement au filtre de Kalman. Là encore, ces filtres ont été déclinés sous diverses variantes (Andrieu et al., 2001; Giremus et al., 2004; Changjiang et al., 2005). Celles-ci permettent par exemple d'adapter le formalisme du filtrage particulaire à des applications de suivi multi-cibles, ou d'adapter le processus de filtrage à une structure hiérarchique du vecteur d'état.

Le filtrage particulaire étant un algorithme fortement plébiscité dans la communauté Vision et présentant une étonnante plasticité pour un champ très varié d'applications, nous avons décidé de privilégier son formalisme pour intégrer une cohérence spatio-temporelle entre nos différentes détections. Cependant, un des points faibles de ce type de filtre est son coût en calcul. En effet, la complexité de cet algorithme croît exponentiellement avec la dimension du vecteur d'état, qui évolue linéairement avec le nombre de particules tandis que l'erreur décroît de manière asymptotique. Cet effet nécessite une augmentation du nombre de particules pour les mêmes performances de filtrage. On obtient souvent un filtre fonctionnant avec un nombre de particules avoisinant 10^4 pour un vecteur d'état de dimension inférieure à 10. De plus, le formalisme Bayésien peut devenir limitant pour la création d'une fonction d'observation efficace.

2.2.2 Optimisation

Dans cette section, nous présentons quelques algorithmes en général utilisés en optimisation, qui sont notamment employés pour le filtrage.

L'algorithme d'Optimisation par Essaim de Particules plus connu sous le nom de PSO (pour Particle Swarm Optimization), a été présenté pour la première fois par (Kennedy and Eberhart, 1995). C'est un algorithme d'optimisation en espace continu. Il est composé d'un nuage de particules communiquant entre elles à la manière des bancs de poissons. Le PSO dérivant des meta-heuristiques, il n'existe pas de preuve quant à sa convergence. D'excellentes performances en vitesse de convergence et une faible erreur résiduelle font de cet algorithme une alternative intéressante dans un grand nombre de domaines tels que la robotique et les réseaux de capteurs (Mei-Ping and Guochang, 2004; Poli, 2008). De plus, il ne nécessite que peu de particules pour une grande efficacité de convergence, ce qui limite fortement son coût calculatoire.

Nous nous sommes donc intéressés de plus près à cet algorithme. En effet, celui-ci fonctionne avec peu de particules, tandis que le filtre particulaire requiert souvent entre 1000 et 10000 particules pour être efficace. Ainsi, une adaptation du PSO en algorithme de suivi nous a semblé opportune.

2.2.3 Suivi visuel hybride

L'idée d'hybridation des algorithmes d'optimisation en méthodes de suivi a déjà été validée dans un certain nombre de travaux, en particulier en ce qui concerne l'algorithme PSO présenté ci-dessus.

Dans (Sha et al., 2015), les auteurs explorent un algorithme de suivi par optimisation fondée sur la ré-initialisation d'un PSO pour chaque image. Ainsi, plusieurs itérations du PSO classique sont utilisées pour affiner le suivi de chaque image. Cependant les auteurs ne présentent que des résultats qualitatifs sur cet algorithme.

Du fait de ses excellentes performances, le PSO a notamment été intégré à des algorithmes de filtrage particulaire. Par exemple, dans (Zhang et al., 2008), les auteurs présentent un algorithme de filtrage particulaire avec quelques itérations de PSO permettant d'affiner la convergence du filtre. Un an plus tard, les mêmes auteurs présentent une extension au suivi multi-cibles dans (Zhang et al., 2009). Cette fois un modèle d'observation spécial est développé pour prendre en compte les occultations et la présence de deux objets. Ensuite, une étape de raffinage par PSO est ajoutée pour chaque objet cible.

Finalement Chen-Chien et al. et Ching-Han et al. appliquent le PSO originel directement à un problème de suivi visuel (Chen-Chien and Guo-Tang, 2012) et (Ching-Han and Miao-Chun, 2011). Cependant, aucun résultat n'est exposé pour valider l'algorithme de suivi.

Nous nous sommes inspirés de ces travaux pour présenter un algorithme de suivi restant fidèle à l'optimisation par essaim de particules original tout en remédiant aux faiblesses de l'algorithme présenté dans (Chen-Chien and Guo-Tang, 2012) et (Ching-Han and Miao-Chun, 2011). Notre variante du PSO vise la même efficacité que le filtrage particulaire tout en présentant une réduction du coût calculatoire.

2.3 Formalisme

Dans le but d'améliorer les performances des modalités de perception visuelle présentées dans les sections 1.4.1 et 1.4.2 du chapitre 1, nous avons ajouté une étape de filtrage permettant de rajouter une cohérence spatio-temporelle entre les différentes modalités et supprimer les détections incohérentes. En effet, il paraît peu naturel que le visage forme un angle de 180 degrés avec les épaules. Pour cela, nous avons exploré quatre techniques de filtrage différentes. Deux techniques basées sur du filtrage particulaire classique, une technique basée sur une hybridation entre un filtre particulaire et un algorithme d'optimisation par essaim de particules (PSO), et enfin notre variante inspirée directement de l'algorithme PSO original.

2.3.1 Filtrage particulaire

Le filtrage particulaire est une méthode Bayésienne séquentielle permettant d'approcher la distribution d'un processus observé indirectement et/ou de manière bruitée. La méthode repose sur l'exploration de l'espace d'état par un ensemble de N particules échantillonnées de manière aléatoire suivant un modèle de dynamique correspondant au processus à estimer. Les filtres particulaires font l'hypothèse que le processus de filtrage peut être modélisé par l'équation (2.1), où le vecteur $\mathbf{x_t}$ représente le vecteur d'état à l'itération t, et le vecteur $\mathbf{z_t}$ représente l'observation associée. Les vecteurs $\mathbf{x_t}|\mathbf{x_{t-1}}$ forment ainsi une chaîne de Markov du premier ordre.

$$\begin{cases} \mathbf{x_t}|\mathbf{x_{t-1}} \sim p_{\mathbf{x_t}|\mathbf{x_{t-1}}}(\mathbf{x}|\mathbf{x_{t-1}}) \\ \mathbf{z_t}|\mathbf{x_t} \sim p_{\mathbf{z_t}|\mathbf{x_t}}(\mathbf{z}|\mathbf{x_t}) \end{cases} \tag{2.1}$$

Un exemple de ce modèle est représenté par l'équation (2.2). Ici, les vecteurs $\mathbf{v_t}$ et $\mathbf{w_t}$ sont échantillonnés de manière indépendante et suivent les densités de probabilité définies par l'équation (2.1). Les fonctions $f(\cdot)$ et $h(\cdot)$ sont *a priori* connues. Lorsque ces deux fonctions sont linéaires, et où les vecteurs $\mathbf{v_t}$ et $\mathbf{w_t}$ sont échantillonnés suivant des densités gaussiennes, nous retombons dans le cas d'un filtre de Kalman. L'avantage du filtre particulaire réside dans le fait que les modèles de dynamique et d'observation peuvent être non-linéaires et non-gaussiens.

$$\begin{cases} \mathbf{x_t} = f(\mathbf{x_{t-1}}) + \mathbf{v_t} \\ \mathbf{z_t} = h(\mathbf{x_t}) + \mathbf{w_t} \end{cases} \tag{2.2}$$

Dans ce travail, nous nous sommes focalisés sur l'algorithme de filtrage particulaire de type échantillonnage avec ré-échantillonnage par importance (Sampling Importance Resampling : SIR dans la suite). La tendance des filtres particulaires à dégénérer (Kong et al., 1994) constitue une faiblesse, car les particules se répandent dans tout l'espace d'état. De ce fait, un grand nombre de particules peuvent se retrouver affectées par un poids négligeable ne contribuant pas à l'estimation de la densité de filtrage de manière efficace. Pour résoudre ce problème, nous pourrions multiplier le nombre de particules. Cependant, cela ferait exploser le coût calculatoire pour un gain en précision d'estimation proche de zéro. L'étape de ré-échantillonnage a donc été introduite pour regrouper les particules autour des modes de la distribution de filtrage, afin que chaque particule participe au mieux à l'estimation de celle-ci.

Algorithme 1 : Algorithme SIR, (Doucet et al., 2000)

Résultat : $\{(\boldsymbol{x_t^{(i)}}, w_t^{(i)})\}_{i=1}^N = SIR(\{(\boldsymbol{x_{t-1}^{(i)}}, w_{t-1}^{(i)})\}_{i=1}^N, \boldsymbol{z_t})$
if $t = 0$ **then**
  Echantillonner $\boldsymbol{x_0^{(1)}}$,...,$\boldsymbol{x_0^{(N)}}$ i.i.d selon $p_0(\boldsymbol{x_0})$, et poser $w_0^{(i)} = \frac{1}{N}$
else if $t \geq 1$ **then**
  $\{(\boldsymbol{x_{t-1}^{(i)}}, w_{t-1}^{(i)})\}_{i=0}^N$ représente $p(\boldsymbol{x_{t-1}}, \boldsymbol{z_{1:t-1}})$
  for $i = 1,...,N$ **do**
    Echantillonner indépendamment $\boldsymbol{x_t^{(i)}} \sim q(\boldsymbol{x_t^{(i)}}|\boldsymbol{x_{t-1}^{(i)}}, \boldsymbol{z_t})$
    Mettre à jour les poids via $w_t^{(i)} \propto \frac{p(\boldsymbol{z_t}|\boldsymbol{x_t^{(i)}})p(\boldsymbol{x_t^{(i)}}|\boldsymbol{x_{t-1}^{(i)}})}{q(\boldsymbol{x_t^{(i)}}|\boldsymbol{x_{t-1}^{(i)}}, \boldsymbol{z_t})}$
  Normaliser les poids de sorte que $\sum_i w_t^{(i)} = 1$
  Calculer l'estimée MMSE $E_{p(\boldsymbol{x_t}|\boldsymbol{x_{t-1}})}[\boldsymbol{x_t}] = \sum_{i=1}^N w_t^{(i)} \boldsymbol{x_t^{(i)}}$
  Rééchantillonner $\{(\boldsymbol{x_t^{(i)}}, w_t^{(i)})\}_{i=1}^N$ en $\{(\boldsymbol{x_t^{(i)}}, \frac{1}{N})\}_{i=1}^N$ à l'aide de l'algorithme 2.

Algorithme 2 : Algorithme RESAMPLE

Résultat : $\{(\boldsymbol{x_t^{(s(i))}}, w_t^{(s(i))})\}_{i=1}^N = RESAMPLE(\{(\boldsymbol{x_t^{(i)}}, w_t^{(i)})\}_{i=1}^N)$
Initialiser la Somme Cumulée des Poids (SCP) $c_1 = w_t^{(1)}$
for $i = 2,...,N$ **do**
    Construire la SCP $c_i = c_{i-1} + w_t^{(i)}$
Poser $i = 1$
Echantillonner un point de départ : $u_1 \sim \mathcal{U}_{[0,N^{-1}]}$
for $j = 1,...,N$ **do**
    Se déplacer le long de la SCP : $u_j = u_1 + (j-1)N^{-1}$
    while $u_j > c_i$ **do**
        $i = i + 1$
    Recopier la particule $\boldsymbol{x_t^{(s(j))}} = \boldsymbol{x_t^{(i)}}$
    Affecter le poids : $w_t^{(s(j))} = N^{-1}$

Les implémentations du SIR et l'étape de ré-échantillonnage sont respectivement détaillées dans les algorithmes 1 et 2. $\mathbf{x_t^{(i)}}$ représente le vecteur d'état de la $i^{ème}$ particule à l'instant t. $w_t^{(i)}$ représente le poids associé à la $i^{ème}$ particule à l'instant t. $p_0(\cdot)$ est la densité *a priori* de probabilité de l'état du système. $p(\cdot)$ et $q(\cdot)$ représentent respectivement la densité du système à un instant donné et la fonction d'importance.

L'étape d'initialisation de l'algorithme consiste à échantillonner l'ensemble du nuage de particules selon la loi de densité *a priori* du système. En général, nous choisissons la loi de manière à couvrir au mieux l'espace d'état de notre système. L'ensemble du nuage de particules est initialisé avec des poids uniformes. Après cette étape, on ré-échantillonne les particules suivant la fonction d'importance $q(\cdot)$. Si nous prenons $q(\cdot) = p(\cdot)$, nous pouvons simplifier la mise à jour des poids. Cette étape devient alors proportionnelle à la vraisemblance du système i.e. la fonction d'observation. Cette simplification transforme l'algorithme SIR en algorithme CONDENSATION, introduit dans (Isard and Blake, 1998). La fonction d'observation est donc évaluée pour chaque particule, ce qui donne le poids associé. C'est en général la partie demandant le plus de temps de calcul dans cet algorithme, et donc la partie qui détermine si le suivi s'effectue en temps réel ou non. L'étape suivante consiste à normaliser la somme des poids.

Finalement, une dernière étape de ré-échantillonnage par importance est ajoutée. Celle-ci permet de dupliquer les particules ayant le poids w_i le plus important tandis que les particules avec le poids le plus faible seront détruites. Nous évitons ainsi une dégénérescence du système tout en ayant une bonne représentation des principaux modes de la distribution. Cependant, cette étape a tendance à supprimer les modes secondaires les plus faibles, ce qui peut poser problème dans certaines applications. Une mesure du *nombre de particules efficaces* (N_{eff}) existe dans le formalisme du filtrage particulaire. Cette mesure permet de connaître le nombre de particules contribuant effectivement à l'estimation de la densité de filtrage. Elle est rappelée dans l'équation (2.3). Plus

la valeur de N_{eff} est élevée et tend vers N (nombre de particules), moins le nuage de particules dégénère. Certains algorithmes conditionnent l'étape de ré-échantillonnage par un test sur la valeur du paramètre N_{eff}.

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} w_i^2} \tag{2.3}$$

Une illustration de la transition entre deux itérations de cet algorithme est représentée dans la figure 2.2. L'étape d'échantillonnage, où la position des particules est tirée aléatoirement selon la dynamique du système, est représentée en vert. Les particules sont ensuite pondérées suivant la loi d'observation en bleu. L'étape de ré-échantillonnage par importance (en orange) permet de dupliquer les particules ayant le plus de poids dans la représentation de la densité de filtrage. Elles sont ensuite re-propagées à l'itération suivante selon la dynamique du système. Les courbes bleues représentent la vraisemblance projetée dans une dimension à chaque instant t.

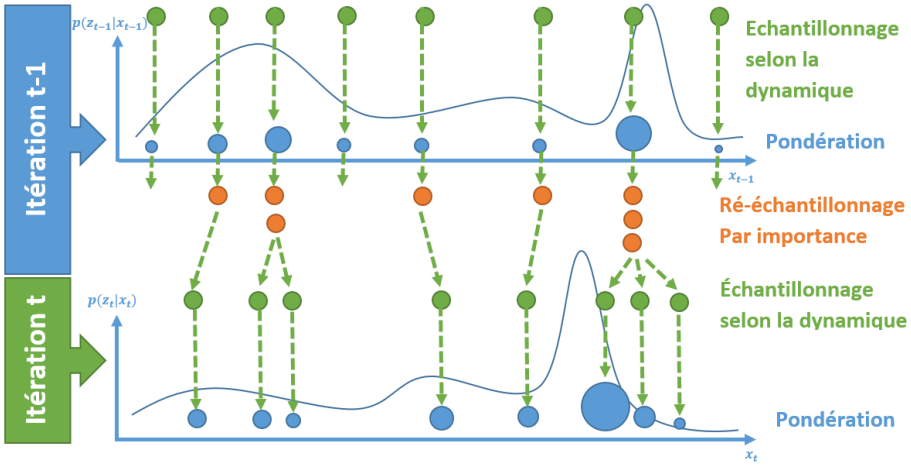


FIGURE 2.2 – Schématisation du processus de suivi par filtrage particulaire.

Dans notre contexte applicatif, nous avons choisi deux modèles de dynamique. Le premier que nous détaillons ici, est le modèle classique de marche aléatoire gaussienne, tandis que le deuxième est un modèle à vitesse constante.

Marche aléatoire : Pour ce modèle, nous posons $p(\cdot) = q(\cdot)$ de manière à simplifier l'étape de calcul des poids. Nous définissons par l'équation (2.4) la loi de transition comme une marche aléatoire Gaussienne (Random Walk : SIR_RW dans la suite) de moyenne $\boldsymbol{x_{t-1}^{(i)}}$ et de covariance $\boldsymbol{\Sigma}$. Nous nous retrouvons ainsi dans le cas du modèle de dynamique le plus simple.

$$p(\boldsymbol{x_t^{(i)}}|\boldsymbol{x_{t-1}^{(i)}}, \boldsymbol{z_t}) \sim \mathcal{N}(\boldsymbol{x_{t-1}^{(i)}}, \boldsymbol{\Sigma}) \tag{2.4}$$

Vitesse constante : De façon similaire au modèle de marche aléatoire gaussienne, nous posons aussi $p(\cdot) = q(\cdot)$ pour la simplification. La subtilité du modèle à vitesse constante (Constant Velocity : SIR_CV dans la suite) réside dans la construction du vecteur d'état. Par exemple, dans

un filtrage à trois dimensions, notre vecteur d'état serait $\boldsymbol{s} = [x, y, z]$. Dans le modèle de marche aléatoire gaussienne, nous aurions pris $\boldsymbol{x} = \boldsymbol{s}$. Dans ce modèle, nous prenons $\boldsymbol{x} = [\boldsymbol{s}, \dot{\boldsymbol{s}}]$, où $\dot{\boldsymbol{s}}$ représente le vecteur vitesse pour chaque composante du vecteur d'état $\boldsymbol{s}$. La loi de transition va ainsi être définie par l'équation (2.5), où Δ est le facteur de pondération de la vitesse (fixé en fonction du nombre d'images par seconde et de la vitesse apparente de la cible), $\boldsymbol{\Sigma_s}$ est la matrice de covariance de la première partie $\boldsymbol{s}$ de notre vecteur d'état, et $\boldsymbol{\Sigma_{\dot{s}}}$ est la matrice de covariance de $\dot{\boldsymbol{s}}$.

$$p(\boldsymbol{x_t^{(i)}}|\boldsymbol{x_{t-1}^{(i)}}, \boldsymbol{z_t}) \sim [\mathcal{N}(\boldsymbol{s_{t-1}^{(i)}} + \Delta\dot{\boldsymbol{s}}_{\boldsymbol{t-1}}^{\boldsymbol{(i)}}, \boldsymbol{\Sigma}_s), \mathcal{N}(\dot{\boldsymbol{s}}_{\boldsymbol{t-1}}^{\boldsymbol{(i)}}, \boldsymbol{\Sigma}_{\dot{s}})] \tag{2.5}$$

Cet algorithme permet de suivre de manière efficace le mouvement de cibles ayant des modèles de dynamique non-linéaires, ce qui est *a priori* le cas des mouvements humains. Cependant, pour que l'algorithme fonctionne correctement il requiert souvent un nombre de particules situé entre $N = 1000$ et $N = 10000$, ce qui peut être une limitation dans le cas des applications embarquées et/ou temps réel.

2.3.2 Optimisation par essaim de particules

L'optimisation par essaim de particules (ou Particle Swarm Optimization : PSO dans la suite) est un algorithme dérivant des algorithmes génétiques. Il appartient à la branche des meta-heuristiques, il n'y a alors aucune preuve de convergence. Cependant, cet algorithme a prouvé son efficacité et se voit de plus en plus utilisé dans un grand nombre d'applications, notamment la vision par ordinateur, l'énergie et la finance (Mei-Ping and Guo-chang, 2004; Poli, 2008). Le PSO que nous décrivons dans l'algorithme 3, est inspiré des mouvements de murmure d'oiseaux et de bancs de poissons. Un grand nombre de particules (oiseaux) va explorer l'espace d'état du système afin de converger vers un maximum global (nourriture). Ce qui rend cet algorithme extrêmement efficace est le fait que chaque particule communique avec les autres, elles ne sont donc pas indépendantes à la manière d'un filtrage particulaire. Une particule $\boldsymbol{x^{(i)}}$ va trouver le point de l'espace d'état le plus intéressant pour elle (maximum local) $\boldsymbol{p^{(i)}}$, puis le comparer avec l'ensemble des maxima de chaque particule. Un maximum global $\boldsymbol{g}$ présentant le meilleur score est alors trouvé, ce qui permettra d'orienter le nuage de particules dans sa direction. Contrairement à la section précédente, nous ne travaillons pas en temporel (indice t) avec le PSO mais en itérations (indice k).

La $i^{ème}$ particule à l'itération k est représentée par le vecteur d'état $\boldsymbol{x_k^{(i)}}$ et sa vitesse dans l'espace d'état par $\boldsymbol{v_k^{(i)}}$. Trois poids permettent de paramétrer le comportement du nuage de particules : la composante **inertielle**, la composante **cognitive** et la composante **sociale**. Elles sont notées respectivement ω, ψ_p et ψ_g.

Le premier poids ω permet de régler la composante inertielle de la particule. Ainsi, plus ce poids est grand, plus il est difficile de changer la trajectoire de la particule. Une inertie trop importante peut alors conduire à l'oscillation, voire la divergence du nuage de particules. Cependant, une inertie trop faible risque de faire converger le nuage vers un maximum local.

Le deuxième paramètre ψ_p est la composante cognitive des particules. Elle représente la mémoire de la particule, i.e. le dernier point ayant le plus haut score (maximum local) exploré par la particule

Algorithme 3 : Algorithme PSO

Résultat : Estimateur MAP : $\hat{\boldsymbol{x}} = \boldsymbol{g}$
Initialisation : **for** $i = 1$ *à* N **do**
    $\boldsymbol{x_0^{(i)}} \sim \mathcal{U}_{[b_{lo}, b_{up}]}$
    $\boldsymbol{p_0^{(i)}} \leftarrow \boldsymbol{x_0^{(i)}}$, $\boldsymbol{g_0} \leftarrow argmax(f(\boldsymbol{p_0^{(i)}}))$
    $\boldsymbol{v_0^{(i)}} \sim \mathcal{U}_{[-|b_{up}-b_{lo}|, |b_{up}-b_{lo}|]}$
while *($k < K_{max}$) et ($\epsilon > S$)* **do**
    for $i = 1$ *à* N **do**
        r_p, $r_g \sim \mathcal{U}_{[0,1]}$
        $\boldsymbol{v_k^{(i)}} \leftarrow \omega \boldsymbol{v_{k-1}^{(i)}} + \psi_p r_p (\boldsymbol{p_{k-1}^{(i)}} - \boldsymbol{x_{k-1}^{(i)}}) + \psi_g r_g (\boldsymbol{g_{k-1}} - \boldsymbol{x_{k-1}^{(i)}})$
        $\boldsymbol{x_k^{(i)}} \leftarrow \boldsymbol{x_{k-1}^{(i)}} + \boldsymbol{v_{k-1}^{(i)}}$
        if $f(\boldsymbol{x_k^{(i)}}) > f(\boldsymbol{p_{k-1}^{(i)}})$ **then**
            $\boldsymbol{p_k^{(i)}} \leftarrow \boldsymbol{x_k^{(i)}}$
    $\boldsymbol{g_k} \leftarrow argmax(f(\boldsymbol{p_k^{(i)}}))$

$\boldsymbol{p_k^{(i)}}$. Ainsi, si la particule s'éloigne trop sans trouver de nouveau maximum local, elle a tendance à revenir vers ce point.

Enfin le dernier paramètre ψ_g correspond à la composante sociale. Il pondère le maximum global $\boldsymbol{g_k}$ trouvé par l'ensemble du nuage de particules. Cette composante a tendance à orienter le nuage vers le maximum global de l'espace d'état.

Les deux derniers paramètres doivent être fixés l'un par rapport à l'autre puisqu'un ψ_g trop élevé risquerait d'entraîner une convergence rapide vers un maximum local. Cependant, un ψ_p trop élevé risque d'allonger dramatiquement le temps de convergence. Les paramètres r_p et r_g sont choisis aléatoirement à chaque itération pour pondérer la composante cognitive et la composante sociale. Cela réduit le risque de dysfonctionnement en cas de réglage inadapté des paramètres ψ_p et ψ_g. De plus, pour une itération, les deux paramètres r_p et r_g peuvent être suffisamment faibles pour que seule l'inertie de la particule soit active. b_{up} et b_{lo} représentent respectivement les limites supérieure et inférieure de l'espace d'état. Le critère d'arrêt est défini soit par un nombre d'itérations maximal K_{max}, soit par un seuil S de l'erreur résiduelle ϵ. Le vecteur d'état $\boldsymbol{g}$ renvoyé par l'algorithme pouvant être assimilé au Maximum A Posteriori (MAP). Le maximum est évalué par une fonction $f(\cdot)$ permettant de calculer le score de chaque position dans l'espace d'état. L'analogie peut être faite avec l'étape de calcul de la vraisemblance présentée dans le filtrage particulaire dans la section 2.3.1. Dans cet algorithme cette étape est également la plus coûteuse en temps de calcul.

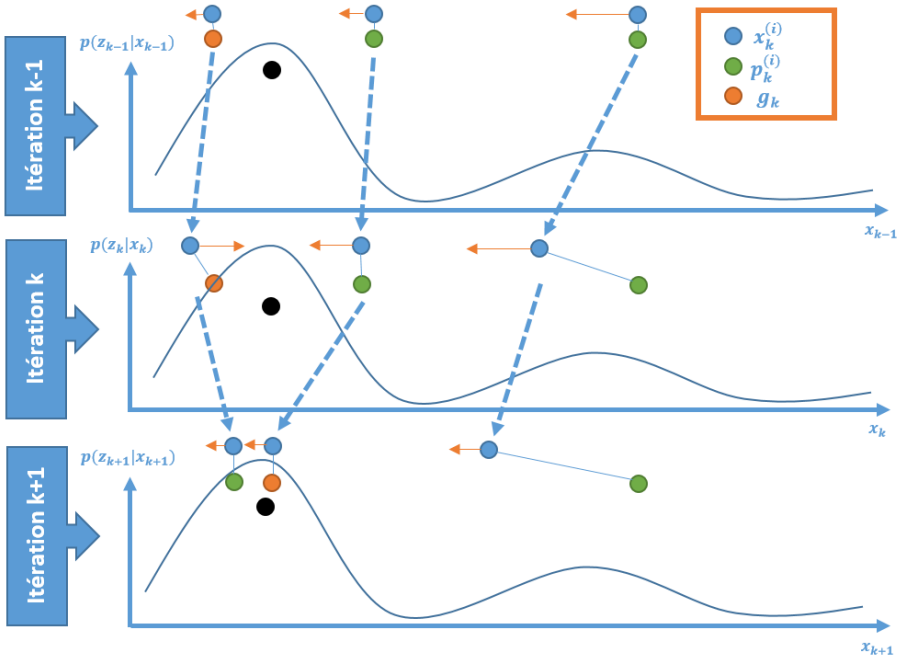


FIGURE 2.3 – Schématisation du processus d'optimisation par essaim de particules (PSO).

Le fonctionnement de cet algorithme d'optimisation est illustré sur la figure 2.3. Sur ce schéma, nous avons représenté trois itérations du PSO. Les ronds bleus représentent les particules se déplaçant dans l'espace d'état. Les flèches orange représentent la vitesse de chaque particule, chacune étant associée à un maximum local $\boldsymbol{p_k^{(i)}}$ représenté en vert. Lorsque ce maximum local se trouve être le maximum global, il devient $\boldsymbol{g_k}$ et est représenté en orange. La courbe représente la vraisemblance projetée dans une dimension à chaque itération. Dans un contexte d'optimisation, celle-ci n'est pas modifiée d'une itération à l'autre.

2.3.3 Stratégie hybride : optimisation séquentielle par essaim de particules

La stratégie hybride d'optimisation séquentielle par essaim de particules (SPSO) que nous présentons ici a été initiée par Zhang et al. dans (Zhang et al., 2008). C'est une hybridation entre un algorithme de filtrage particulaire de type SIR et un algorithme d'optimisation de type PSO. En effet, le filtrage est effectué exactement comme un filtrage particulaire classique. Cependant, une étape d'optimisation supplémentaire permet d'affiner le résultat de l'itération de filtrage. Cette modification du SIR initial ressemble à une méthode d'échantillonnage par importance multi-couche (ou Multi-Layer Importance Sampling).

Ainsi, dans l'algorithme 4, la $i^{ème}$ particule $\boldsymbol{x_t^{(i)}}$ est propagée à l'instant t suivant une densité gaussienne de covariance $\boldsymbol{\Sigma}$ et de moyenne $\boldsymbol{p_t^{(i)}}$. $\boldsymbol{p_t^{(i)}}$ représente le maximum local trouvé par la

Algorithme 4 : Algorithme SPSO

$\{(\boldsymbol{x_t^{(i)}})\}_{i=1}^N = SPSO(\{(\boldsymbol{x_{t-1}^{(i)}})\}_{i=1}^N, \boldsymbol{z_t})$
Résultat : Estimateur MMSE : $\hat{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{x^{(i)}}$
Initialisation : **if** $t = 0$ **then**
    Echantillonner $\boldsymbol{x_0^{(1)}}$,...,$\boldsymbol{x_0^{(N)}}$ i.i.d selon $p_0(\boldsymbol{x_0})$
    $\boldsymbol{p_0^{(i)}} \leftarrow \boldsymbol{x_0^{(i)}}$, $\boldsymbol{g_0} \leftarrow argmax(f(\boldsymbol{p_0^{(i)}}))$
else if $t \geq 1$ **then**
    for $i = 1,...,N$ **do**
        Echantillonner $\boldsymbol{x_t^{(i)}} \sim N(\boldsymbol{p_t^{(i)}}, \boldsymbol{\Sigma})$
    K_{max} itérations de PSO (algorithme 3)
    Calculer l'estimée MMSE $E_{p(\boldsymbol{x_t}|\boldsymbol{x_{t-1}})}[\boldsymbol{x_t}] = \frac{1}{N}\sum_{i=1}^N \boldsymbol{x_t^{(i)}}$

particule lors des itérations effectuées par l'algorithme PSO décrit dans la sous-section précédente. À l'instar de l'algorithme PSO, nous pouvons remarquer la présence d'une fonction d'évaluation $f(\cdot)$ qui est une fonction de vraisemblance pour pouvoir rester dans le formalisme bayésien. Nous remarquons ainsi une des faiblesses de ce filtre : son coût en calcul. En effet, à chaque itération, nous devons évaluer N fois la fonction de vraisemblance, et pour chaque particule, calculer K_{max} fois la fonction de vraisemblance durant l'étape d'optimisation. Ainsi, contrairement au filtre particulaire dont la complexité est $O(N)$ au regard de la vraisemblance, la complexité du SPSO évolue en $O(N * K_{max})$ où N est le nombre de particules, et K_{max} est le nombre d'itérations de l'étape PSO.

2.3.4 Filtrage par essaim de particules

Dans cette section, nous proposons notre contribution : une nouvelle solution de filtrage essayant d'exploiter au mieux l'efficacité et la rapidité en convergence de l'algorithme PSO initial, tout en gardant sa légèreté en coût de calcul.

Lors d'un processus d'optimisation, l'algorithme parcourt un espace d'état en essayant de trouver un maximum global. Lors d'une séquence de suivi de cible en revanche, l'espace d'état change à chaque itération, tout comme la position et la valeur du maximum global. Il peut aussi parfois disparaître de l'espace d'état durant quelques itérations du fait des sorties de champs. Nous avons trouvé pertinent la transformation du PSO en algorithme de suivi. En effet, les particules ayant une inertie, le nuage devrait naturellement prendre la dynamique de la cible en mouvement. En cas de perte de cible, il pourrait *a priori* extrapoler le mouvement du maximum global pendant quelques itérations.

Suivant ces postulats, nous avons modifié l'algorithme PSO originel pour en faire un algorithme de suivi détaillé dans l'algorithme 5 dénommé PSOT (pour Particle Swarm Optimization inspired Tracker). Ici $\boldsymbol{x_t^{(i)}}$ désigne le vecteur d'état de la $i^{ème}$ particule à l'instant t et $\boldsymbol{v_t^{(i)}}$ est la vitesse associée. Les paramètres ω, ψ_p, et ψ_g restent les mêmes que ceux utilisés dans l'algorithme 3 avec

les mêmes effets sur le comportement du nuage de particules.

Le principal changement à apporter à l'algorithme concerne l'évolution de la position de la cible à suivre. En effet, le seul moment où l'algorithme interagit avec l'espace d'état se situant durant le calcul de la fonction d'évaluation $f(\cdot)$. Ainsi, le dernier maximum local connu de la particule $\boldsymbol{p_t^{(i)}}$ est susceptible d'avoir changé de place depuis l'itération précédente. Pour prendre cela en compte, nous essayons de prédire la nouvelle position du dernier maximum local grâce à une fonction $d(\cdot)$ qui représente la dynamique du système par analogie avec le filtre particulaire. Ensuite, comme la valeur de $f(\boldsymbol{p_t^{(i)}})$ change avec l'évolution de l'espace d'état depuis la dernière itération, il faut la mettre à jour et donc à nouveau calculer la fonction d'évaluation. Cela vient en contradiction avec les travaux de Chen-Chien et al. (Chen-Chien and Guo-Tang, 2012). En effet, en utilisant le PSO initial tel quel, nous avons remarqué que l'algorithme finit invariablement par se bloquer : lorsque le score de la fonction d'évaluation atteint une valeur trop importante, plus aucune valeur ne peut prendre sa place comme maximum global : l'algorithme a alors convergé. Ces deux modifications permettent d'obtenir un algorithme de suivi robuste aux changements brusques de dynamique du système et d'éviter sa convergence prématurée. Dans notre application, la fonction $d(\cdot)$ est implémentée suivant l'équation (2.6), où $\boldsymbol{w_t^{(i)}}$ est un vecteur échantillonné aléatoirement et de manière indépendante à chaque instant t.

$$d(\boldsymbol{p_t^{(i)}}) = \boldsymbol{p_t^{(i)}} + \boldsymbol{w_t^{(i)}} \tag{2.6}$$

Contrairement à un algorithme SIR, et de façon similaire au SPSO, notre algorithme PSOT n'entraîne pas de dégénérescence du nuage de particules. Il ne nécessite donc pas d'étape de ré-échantillonnage. Pour vérifier cela, nous avons créé une mesure similaire au paramètre N_{eff} du filtre SIR décrite par l'équation (2.7).

$$N_{eff} = \frac{(\sum_{i=1}^{N} f(x_p^{(i)}))^2}{\sum_{i=1}^{N} f(x_p^{(i)})^2} \tag{2.7}$$

2.4 Évaluations et résultats

Dans cette section, nous présentons les expériences que nous avons menées de façon à valider notre algorithme de suivi par essaim de particules. Nous commençons par exposer les évaluations et résultats à partir de données simulées, puis à partir de données vidéos réelles dans notre contexte applicatif. Nous comparons les performances de notre algorithme PSOT à la technique de filtrage hybride présentée plus haut (SPSO), puis aux algorithmes SIR avec le modèle à vitesse constante (SIR_CV) et le modèle à marche aléatoire gaussienne (SIR_RW).

2.4.1 Évaluations sur signaux synthétiques

Nous avons tout d'abord validé la faisabilité du filtre en simulant un certain nombre de trajectoires. Nous avons ainsi pu observer le comportement des différents filtres présentés dans la section

Algorithme 5 : PSOT

$\{(\boldsymbol{x}_t^{(i)})\}_{i=1}^N = PSOT(\{(\boldsymbol{x}_{t-1}^{(i)})\}_{i=1}^N, \boldsymbol{z}_t)$
Résultat : Estimateur MAP ou MMSE pour chaque image
for *i=1 to N (initialisation)* **do**
  $\boldsymbol{x}_0^{(i)} \sim U(b_{lo}, b_{up})$
  $\boldsymbol{p}_0^{(i)} \leftarrow \boldsymbol{x}_0^{(i)}$, $\boldsymbol{g}_0 \leftarrow argmax(f(\boldsymbol{p}_0^{(i)}))$
  $\boldsymbol{v}_0^{(i)} \sim U(-|b_{up} - b_{lo}|, |b_{up} - b_{lo}|)$
if $t \geq 1$ **then**
  for *i=1 to N* **do**
    r_p, $r_g \sim U(0,1)$
    $\boldsymbol{v}_t^{(i)} \leftarrow \omega \boldsymbol{v}_{t-1}^{(i)} + \psi_p r_p (d(\boldsymbol{p}_{t-1}^{(i)}) - \boldsymbol{x}_{t-1}^{(i)}) + \psi_g r_g (d(\boldsymbol{g}_{t-1}) - \boldsymbol{x}_{t-1}^{(i)})$
    $\boldsymbol{x}_{t-1}^{(i)} \leftarrow \boldsymbol{x}_{t-1}^{(i)} + \boldsymbol{v}_t^{(i)}$
    if $f(\boldsymbol{x}_t^{(i)}) > f(\boldsymbol{p}_{t-1}^{(i)})$ **then**
      $\boldsymbol{p}_t^{(i)} \leftarrow \boldsymbol{x}_t^{(i)}$
  $\boldsymbol{g}_t \leftarrow argmax(f(\boldsymbol{p}_t^{(i)}))$
  MAP estimator : $\hat{\boldsymbol{x}}_t = \boldsymbol{g}_t$ ou
  MMSE estimator : $\hat{\boldsymbol{x}}_t = \sum_{i=1}^N \frac{f(\boldsymbol{p}_t^{(i)})}{\sum_{i=0}^N f(\boldsymbol{p}_t^{(i)})} \boldsymbol{x}_t^{(i)}$

2.3. La première étape a consisté à faire varier différents paramètres tels que le nombre de particules, le nombre de dimensions et le rapport signal sur bruit. Le comportement du filtre en simulation a ainsi pu être validé. Cela a permis de vérifier son fonctionnement en environnement maîtrisé.

Pour cette évaluation, nous avons égalisé tous les paramètres ayant le même effet dans les quatre filtres présentés :

- le nombre de particules N et la covariance Σ pour les modèles de dynamique des quatre algorithmes.
- ω pour le SPSO, le PSOT, et le SIR avec modèle de dynamique à vitesse constante (SIR_CV).
- ψ_p et ψ_g pour le SPSO et le PSOT.

Sur la figure 2.4, nous pouvons observer les différents résultats de simulation. La racine de l'erreur quadratique moyenne est ainsi affichée pour 50 simulations. Pour l'algorithme PSOT, on affiche à la fois l'estimateur du Maximum A Posteriori (MAP), et l'estimateur du minimum d'erreur quadratique moyenne (ou Minimum Mean Square Error : MMSE). Nous observons alors plusieurs comportements.

Nombre de particules : premièrement, sur la figure 2.4a, nous pouvons remarquer que l'erreur croît lorsque le nombre de particules diminue, en particulier pour les filtres SIR. Nous observons que pour 500 particules, nous obtenons les meilleurs résultats avec le PSOT. Or, plus le nombre de particules est faible, moins l'algorithme est coûteux en calcul, augmentant donc sa rapidité

d'exécution, tout cela étant un critère important pour une application temps réel.

Dimension : sur la figure 2.4b, nous pouvons voir que la précision diminue lorsque la dimension augmente. Ici également, les algorithmes PSOT et SPSO semblent être les plus robustes à ce changement de dimension.

Bruit : sur la figure 2.4c, nous remarquons que le SPSO et PSOT demeurent les plus précis lorsque le rapport signal sur bruit (signal to noise ratio : SNR) augmente. Le SNR est défini par l'équation (2.8). Plus celui-ci est élevé, moins le signal est bruité. Cette robustesse au bruit est un paramètre important des algorithmes de filtrage puisqu'elle réduit la sensibilité de l'algorithme aux données incohérentes. De plus, dans notre contexte robotique, le bruit est présent à tous les niveaux : au niveau sonore avec le bruit du robot, mais aussi au niveau visuel avec les changements d'illuminations et les fausses détections fréquentes dues à un environnement humain complexe à traiter pour les algorithmes de vision.

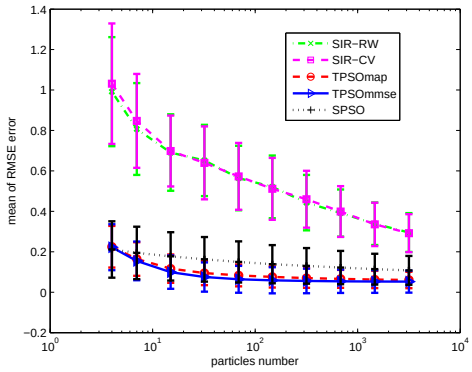
$$SNR = 10log(\frac{PuissanceSignal}{PuissanceBruit}) \tag{2.8}$$

Nombre de particules efficaces : finalement, la figure 2.4d illustre la discussion abordée dans la section 2.3.4. Nous remarquons que les filtres donnant les meilleurs résultats sont en général ceux ayant le N_{eff} le plus élevé, c'est-à-dire, ceux qui ne dégénèrent pas. Nous pouvons voir sur cette figure que le filtre SIR avec modèle de dynamique à vitesse constante a moins tendance à dégénérer que le filtre SIR avec marche aléatoire. Nous pouvons aussi voir que notre algorithme PSOT est le filtre qui a le nombre de particules efficaces (N_{eff}) le plus élevé. Ceci peut donc expliquer la meilleure précision que nous obtenons pour les simulations précédentes.

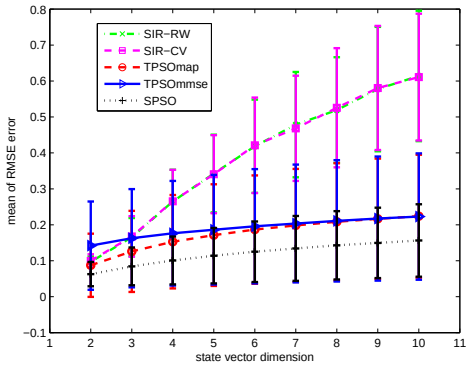
Ces expériences ont permis de caractériser le comportement du filtre par essaim de particules confronté à différents types de perturbations. Ces résultats préliminaires placent cet algorithme de filtrage comme une alternative intéressante et moins coûteuse en temps de calcul vis-à-vis des filtres particulaires les plus simples.

2.4.2 Évaluations sur données visuelles

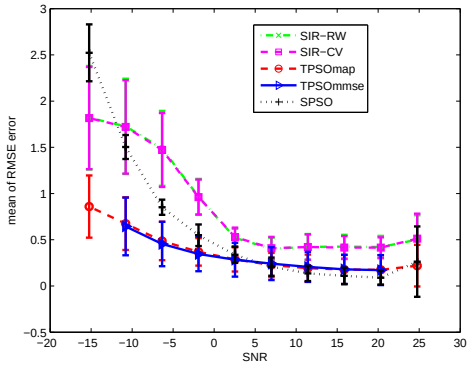
L'évaluation de l'algorithme de filtrage en situation réelle a été effectuée à partir d'un flux vidéo d'environ quatre minutes avec une vérité terrain de 30 images par seconde. Pour cela, nous avons fait porter un casque à la personne suivie, celui-ci représentant l'orientation du visage et la position de la tête, ainsi qu'une tige dans le dos pour marquer l'orientation des épaules (voir la figure 2.6). Toute l'expérience a été enregistrée avec un capteur RGB-D calibré par rapport au repère monde pour pouvoir comparer les résultats de filtrage par rapport à la vérité terrain acquise avec un système industriel de capture de mouvement.



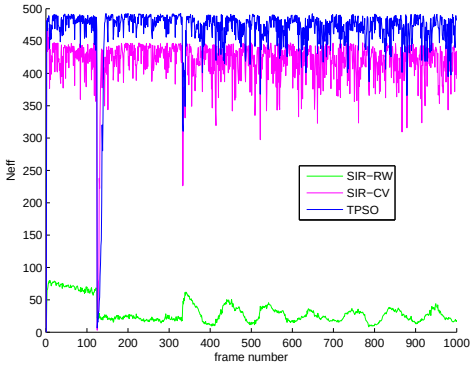
(a) Nombre de particules



(b) Dimension du vecteur d'état



(c) Rapport signal sur bruit (SNR)



(d) N_{eff} moyen sur une séquence

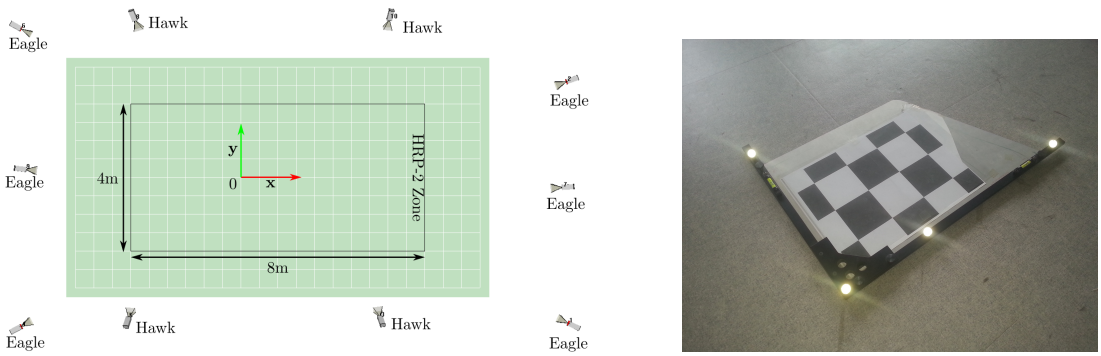
FIGURE 2.4 – Résultats de simulation pour les filtres SIR-RW, SIR-CV, SPSO et PSOT.

Celui-ci a été conçu par la société Motion Analysis [1] et comporte un ensemble de caméras infrarouges (voir la figure 2.5a), de marqueurs réfléchissants, et un logiciel d'acquisition. Il permet ainsi de récupérer la position 3D d'objets repérés par un ensemble de marqueurs dans le repère monde dont l'origine se trouve au niveau du sol. Cependant, pour une estimation efficace de ces positions, ce dispositif requiert un étalonnage de l'espace où sont réalisées les acquisitions.

Pour ces expériences, nous avons donc calibré le capteur Kinect pour pouvoir estimer la matrice permettant de passer du repère Kinect au repère du Motion Capture. Cette estimation est faite à l'aide d'une mire (voir la figure 2.5b comportant à la fois un damier (pour la détection par le capteur Kinect) et des marqueurs (pour la capture de mouvements).

Pour cette application, le vecteur d'état représentant notre système est un vecteur à sept dimensions : trois dimensions pour les coordonnées x, y et z du visage, trois dimensions pour les angles α, β et γ représentant son orientation, et enfin une dimension pour l'angle θ représentant l'orien-

1. http ://www.motionanalysis.com/



(a) Disposition des caméras.

(b) Mire d'étalonnage.

FIGURE 2.5 – Dispositif de capture de mouvement et répartition des caméras dans la salle Gérard Bausil du LAAS-CNRS, et mire d'étalonnage.

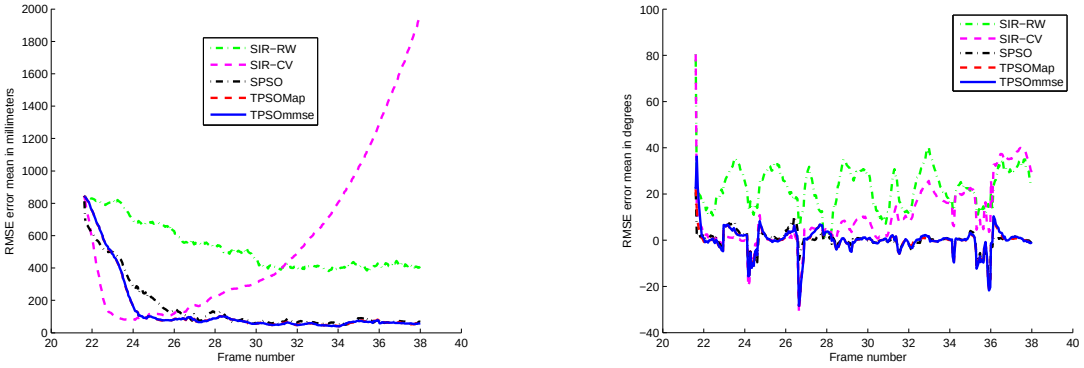
tation des épaules. Nous obtenons alors le vecteur d'état suivant $\boldsymbol{x} = [x\ y\ z\ \alpha\ \beta\ \gamma\ \theta]$. Cependant, une difficulté persiste au niveau du filtrage des angles. En effet, ceux-ci variant entre 180 et -180 degrés, l'évaluation de la vraisemblance est délicate. Ainsi, plutôt que l'échantillonnage direct des angles, nous avons échantillonné le cosinus et le sinus de chaque angle, en respectant la contrainte $cos^2 + sin^2 = 1$. Cela nous permet de garder les angles entre -180 et 180 degrés sans avoir à gérer de discontinuité. Le filtrage est alors effectué sur un vecteur d'état de dimension 11. Pour le filtre particulaire avec modèle à vitesse constante, nous passons à un vecteur d'état de dimension 22, puisqu'il est doublé par les composantes de vitesse.

FIGURE 2.6 – Séquence capturée par le capteur Kinect lors de la construction des bases de données.

De manière analogue, nous avons pour cette expérience, procédé au réglage des paramètres comme pour les simulations. Ces réglages sont résumés dans la table 2.1. Pour chaque filtre, chaque évaluation a été itérée 100 fois. En effet les différents filtres étant de nature stochastique, la répétition du processus permet d'extraire le comportement moyen du filtrage.

TABLE 2.1 – Paramètres utilisés pour l'évaluation du filtre PSOT.

ω	ψ_p	ψ_g	N
0.9	0.8	1	100



(a) RMSE moyenne en position

(b) RMSE moyenne angulaire

FIGURE 2.7 – Résultats de suivi

La figure 2.7a représente l'évolution de l'erreur moyenne RMSE en position exprimée en millimètres sur une portion de la base de données. La figure 2.7b représente également l'évolution de l'erreur moyenne RMSE angulaire en degrés. Nous remarquons que notre filtre obtient les meilleures performances pour un faible nombre de particules.

2.5 Conclusion

Ces expériences ont montré que notre filtre constitue une alternative aux techniques classiques de filtrage particulaire. Non seulement notre solution se trouve être plus robuste au bruit et à l'augmentation de la dimension du vecteur d'état, mais elle converge également avec un nombre de particules 10 à 100 fois plus faible qu'un filtre SIR classique.

Ainsi, si nous considérons la complexité du filtre particulaire comme étant proportionnelle au nombre de particules $O(N)$, celle-ci devrait être $O(2N)$ car nous calculons deux fois la fonction d'évaluation, N représentant le nombre de particules. Cependant, comme mentionné précédemment, notre filtre nécessite 10 à 100 fois moins de particules pour les mêmes performances, ce qui nous permet d'obtenir une complexité relative de $O(2N/10)$ à $O(2N/100)$. Cela rend notre algorithme utilisable pour des applications temps réel nécessitant un faible coût en calcul.

De plus, contrairement au formalisme bayésien, la fonction d'évaluation de chaque particule peut être configurée de façon plus libre que les filtres particulaires. Notre solution est donc à la fois plus robuste et plus flexible tout en étant très facile à implémenter.

L'ensemble de ces contributions a donné lieu à une publication dans la conférence internationale IEEE-ICIP (Mollaret et al., 2014).

Chapitre 3

Contexte et amélioration de l'interaction par perception multimodale

3.1 Introduction

L'interaction homme-robot est une problématique centrale de la robotique d'assistance. Elle est très complexe et difficile à conceptualiser de par le nombre de paramètres à prendre en compte dans sa mise en œuvre. Il faut en outre prendre en considération la perception multimodale de l'homme par le robot, qui nécessite une compréhension du message de l'utilisateur. Il s'agit aussi de déceler ses émotions, d'interpréter sa gestuelle, et plus généralement, tout ce qui a trait au non-verbal. Le robot doit ensuite interpréter ces informations afin de décider de l'action à réaliser et formuler une réponse adéquate. En fonction du contexte, cette action peut s'exprimer de manière verbale et/ou non-verbale.

Le processus d'interaction homme-robot peut se décomposer en 3 étapes comme suit :

- **la perception multimodale de l'utilisateur :** elle commence avec l'analyse du signal audio, la détection des événements correspondant à de la parole et la reconnaissance vocale. Elle peut également être complétée par la perception visuelle comprenant la détection et la reconnaissance de gestes.
- **la gestion de l'interaction :** elle consiste à interpréter, séparément ou conjointement, les stimuli sonores et visuels. Celle-ci est ensuite confrontée à la représentation que le robot se fait du monde en général (connaissances statiques) et à une situation courante en particulier (connaissances dynamiques). Gérer l'interaction, consiste à inférer des informations d'un contexte et agir en conséquences.
- **la génération de la réponse verbale et/ou non verbale :** en situation d'interaction multimodale, l'action réalisée en guise de réponse à l'utilisateur consiste en la production d'un message vocal par le biais d'une synthèse vocale et peut s'accompagner d'un indice non verbal (mouvement, pointage, regard, etc.). Le contexte pouvant aussi influencer la perception

de l'homme (bruit, perte d'audition, distance, etc.) un mouvement ou une indication visuelle peuvent être utilisés comme un complément permettant d'améliorer la compréhension de l'usager.

Dans chacune de ces étapes, le contexte peut varier. Ainsi, l'étape de perception de l'utilisateur par le robot peut être dégradée du fait du bruit ambiant qui varie durant l'interaction, le robot pouvant lui-même en être à l'origine. La position relative de l'utilisateur par rapport au robot peut aussi changer pendant et entre chaque session d'interaction, ayant des effets similaires.

Pour réaliser l'étape d'interaction, le robot doit intégrer un module d'interprétation suffisamment fin pour pouvoir comprendre tous les aspects liés à son environnement. Par exemple, ce module doit être capable de remarquer qu'une partie de l'environnement physique du robot à été modifiée (déplacement d'objets). Au niveau conceptuel, il doit également être capable de comprendre que deux synonymes renvoient au même champ sémantique. Il doit ainsi permettre une certaine flexibilité du langage.

Durant la dernière étape, si l'environnement est bruité, la synthèse vocale pourra être inaudible ou mal comprise. Un complément gestuel et/ou visuel sera alors nécessaire. Toutes ces difficultés posent le cadre de ce chapitre. Nous essayons de répondre à ces différentes problématiques en intervenant au niveau de chacune des trois étapes présentées.

Dans le cadre du projet RIDDLE, le scénario distingue la situation de *monitoring* (à distance de l'utilisateur) de la situation d'interaction proximale. Le robot détecte l'intention de communication de l'utilisateur (voir le chapitre 1 sur l'intentionnalité), ce qui a pour effet d'enclencher la phase d'interaction. Ce chapitre se focalise sur cette étape du scénario.

La phase d'interaction proximale est une situation d'interaction entre le robot et l'utilisateur avec une distance dite sociale estimée entre 1 et 2 mètres. La phase d'interaction dépend ensuite de la tâche d'assistance considérée. Dans ce projet, celle-ci consiste à donner la position d'objets égarés par l'utilisateur.

Cette étape d'interaction proximale est conçue à partir d'un système d'interaction homme-robot, dont l'objectif principal est de comprendre ce qui est en rapport avec cette tâche.

L'un des problèmes majeurs des dispositifs de reconnaissance et de synthèse vocale est l'application dans un contexte robotique. En effet, ceux-ci sont régulièrement employés pour des applications où l'utilisateur se trouve très proche du microphone et du haut-parleur, par exemple avec les assistants vocaux téléphoniques Siri© et Google Now©Ṅous sommes donc très loin d'un contexte robotique avec des tours de parole pouvant être fortement bruités, ou pouvant être perturbés par des mots de remplissage (onomatopées en tous genres). Dans ce contexte, l'utilisateur peut aussi se trouver relativement loin des microphones, ce qui dégrade davantage la qualité du signal de parole.

Dans ce chapitre, nous présentons notre système d'interaction permettant d'intégrer les fonctions de communication de base présentées plus haut, centrées sur la tâche de recherche d'objets. Nous présentons ensuite deux solutions employées dans le but d'améliorer la perception de l'utilisateur en situation de parole distante ainsi que des pistes pour l'amélioration de la prise en compte du contexte.

Un état de l'art sur la chaîne de perception est présenté dans la section 3.2. Notre système d'interaction est ensuite détaillé dans la section 3.3. L'amélioration de la chaîne de perception par

fusion multimodale est formalisée dans la section 3.4 et un dispositif de feedback visuel est exposé dans la section 3.5. Des pistes pour l'amélioration de la gestion de l'interaction en fonction du contexte sont enfin envisagées dans la section 3.6. Nous concluons ce chapitre sur l'amélioration de l'interaction homme-robot dans la section 3.7.

3.2 Etat de l'art

Nous présentons ici un état de l'art sur les différents aspects de notre architecture d'interaction. Nous avons étudié les différentes phases de la chaîne de perception de l'homme par le robot. Celle-ci doit commencer par un système de segmentation en zones de parole et de non-parole appelé détection d'activité vocale (dans la section 3.2.1). Les segments de parole sont ensuite transmis à un moteur de reconnaissance vocale pour réaliser la transcription du signal audio en mots. Ceci est étudié dans la section 3.2.2. Enfin, cette chaîne contient un module de gestion de l'interaction étudié dans la section 3.2.4. Nous présentons aussi les contributions scientifiques en matière de fusion multimodale des systèmes de reconnaissance vocale dans la section 3.2.3. Cette dernière permet d'exposer les travaux liés à la fusion qui permettent l'amélioration de la chaîne de perception de l'utilisateur.

3.2.1 Détection d'activité vocale

La détection d'activité vocale est primordiale dans tout système de reconnaissance vocale. Un signal de parole est composé de zones de parole et de non-parole. La segmentation du flux audio est nécessaire pour traiter uniquement les zones du signal contenant de la parole. Un système de détection d'activité vocale consiste donc essentiellement en un algorithme de classification qui segmente le signal en deux classes.

Différentes techniques sont construites sur l'exploitation de l'énergie du signal (Rocha et al., 2014). Elles font partie des techniques de segmentation les plus simples puisqu'elles se basent sur un seuillage de celle-ci. Avec cette technique, les auteurs arrivent à un taux de vrais positifs de 77,6% pour un signal non bruité. C'est notamment l'une des techniques utilisées dans la bibliothèque Open-Source SphinxBase [1]. Celle-ci intègre de plus une méthode de calibration permettant de déterminer automatiquement le seuil à partir du bruit ambiant, ce qui permet de gérer la variation du bruit ambiant d'une expérience à l'autre.

D'autres travaux sont inspirés des méthodes statistiques de détection. Ainsi, les travaux (Choi and Chang, 2012) exploitent le calcul du gradient spectral. Un test d'hypothèse sur les lois conditionnelles *a posteriori* permet d'extraire l'estimateur C-MAP et ainsi de détecter ou non la présence de parole dans le signal audio. Ils obtiennent des probabilités de fausses alarmes variant de 1% à 20% pour un SNR variant de 15dB à 5dB pour différents types d'environnements bruités. Aucun résultat n'est mentionné concernant le taux de vrais positifs.

Certaines techniques se concentrent sur les fréquences présentes dans le signal audio. C'est par exemple le cas de Wang et al. (Wang et al., 2013). Les auteurs décomposent le signal en multiples bandes de fréquences à l'aide de plus de 40 filtres passe-bandes. Ils mesurent ensuite l'énergie

1. http ://cmusphinx.sourceforge.net/

présente dans ces différentes bandes pour estimer l'étalement en fréquences révélant potentiellement la présence de parole. Enfin, ils élaborent aussi un regroupement temporel de manière à obtenir une détection plus précise. Ils obtiennent ainsi plus de 80% de précision pour des SNR allant jusqu'à -3dB.

Certains travaux sont axés sur la multimodalité. Ainsi, dans (Dov et al., 2015), les auteurs utilisent des descripteurs sonores (MFCC et coefficients de Fourier) et des descripteurs visuels (le flot optique des lèvres et une carte de diffusion). Les MFCC (Mel Fourier Cepstral Coefficients) sont des coefficients cepstraux du signal dans l'échelle MEL. L'utilisation de descripteurs vidéos permet de réduire drastiquement l'influence du bruit sur la qualité de la détection. Ces descripteurs sont ensuite utilisés pour entraîner des GMM (Gaussian Mixture Models). La classification est effectuée à l'aide d'un test d'hypothèses probabiliste. Ils obtiennent ainsi plus de 90% de vrais positifs pour des SNR d'environ 10dB.

La détection d'activité vocale est très importante pour segmenter les zones de parole et déclencher un système de reconnaissance vocale. La précision de ces système est en générale fortement dépendante des corpus d'évaluation. Dans notre contexte de robotique bruitée et de détection distante (qui participe de l'intentionnalité) nous avons à la fois besoin d'un système générique pouvant facilement s'insérer dans une chaîne de perception, tout en permettant une classification en ligne des zones de parole et de non-parole. Dans ces travaux, nous nous sommes orientés vers le détecteur de la bibliothèque SphinxBase. Celui-ci a non seulement l'avantage d'être intégrable directement dans l'architecture du moteur PocketSphinx, mais il possède aussi un système d'autocalibration qui est pertinent dans notre contexte où le SNR peut varier d'une expérience à l'autre. Nous avons utilisé ce même détecteur pour la détection d'intentionnalité afin d'élaborer une architecture compatible.

3.2.2 Reconnaissance vocale

La reconnaissance vocale est très souvent utilisée et peut s'avérer capitale dans la chaîne de perception. Dans cette section, nous présentons les grands axes de recherches sur les algorithmes permettant de l'implémenter.

Dans (Fook et al., 2012), les auteurs comparent dix algorithmes de transcription automatique de la parole basés uniquement sur des descripteurs audios, et neuf algorithmes de transcription fonctionnant à l'aide de descripteurs audios-visuels. Ces algorithmes sont appliqués à la reconnaissance du malais, et sont aussi couramment utilisés pour la reconnaissance d'autres langages en présentant des performances similaires. Toutes ces techniques sont basées sur des modèles HMM (Hidden Markov Model) des réseaux de neurones ou encore des MLP (MultiLayer Perceptrons). Les descripteurs audios extraits sont les MFCC ou les LPCC (Linear Prediction Cepstral Coefficients). Les descripteurs visuels employés sont en général des modèles d'apparence actifs (AAM), des descripteurs basés sur l'apparence ou bien des transformations telles que la DCT (Discrete Cosine Transform). Ainsi ils obtiennent jusqu'à 90% de précision pour les systèmes les plus complexes basés sur des descripteurs uniquement audios, et jusqu'à 91% pour les systèmes audio-visuels. L'incorporation des descripteurs visuels entraîne en général une amélioration de la précision supérieure à 5% par rapport à la même technique fondée sur des informations purement sonores.

Dans leurs travaux (Wan et al., 2013), les auteurs utilisent une technique de soustraction spec-

trale multi-bandes pour réduire la présence de bruit dans le signal de parole, ce qui a pour effet d'améliorer l'extraction des descripteurs MFCC. En condition normale, c'est-à-dire avec un SNR de 40dB, ils obtiennent des performances équivalentes à l'état de l'art avec 90% de précision. Les performances se dégradent ensuite lorsque le SNR diminue (ou le bruit augmente). Cependant, leur technique de débruitage permet d'augmenter la précision de 20% par rapport à l'utilisation du système de reconnaissance vocale sans filtrage. Les auteurs montrent que la soustraction spectrale peut grandement améliorer les performances de reconnaissance vocale en milieu bruité (avion, bruit blanc, usine, etc.).

Bien que les systèmes présentés se basent sur une reconnaissance de phonèmes, certains travaux se focalisent sur une reconnaissance syllabique. C'est par exemple le cas des travaux de Can et Artuner (Can and Artuner, 2013), où les auteurs utilisent une variante des réseaux de neurones pour créer un système de reconnaissance dit grand vocabulaire syllabique à partir de descripteurs MFCC classiques. Ils présentent aussi un détecteur de syllabes basé sur le taux de passage à zéro (ZCR) et l'énergie du signal. Ils rapportent ainsi des précisions de 65,6% pour la reconnaissance vocale et 44% pour le détecteur de frontières syllabiques.

Enfin depuis quelques années de nouvelles techniques commencent à émerger avec l'essor de l'apprentissage profond (*deep learning*). Ainsi, les auteurs de (Deng et al., 2013) présentent une des dernières avancées en matière de reconnaissance vocale et notamment les résultats obtenus en utilisant des techniques de *Deep Neural Network* (DNN) au lieu des traditionnels GMM et HMM. Ils obtiennent ainsi entre 5% et 10% d'amélioration par rapport aux systèmes existants, ces résultats étant prometteurs pour le futur de la reconnaissance vocale.

Cependant, la plupart de ces techniques nécessitent un apprentissage sur de très gros corpus pour être efficaces. Dans notre contexte robotique, nous ne disposons pas de tels volumes de langage parlé. Nous proposons donc l'utilisation d'un système OpenSource, dont les modèles acoustiques ont déjà été appris : PocketSphinx. Bien qu'il utilise les modèles du LIUM par défaut (Galliano et al., 2006), la bibliothèque permet d'en entraîner de nouveaux. Elle permet aussi la création de ses propres grammaires. À titre de comparaison, nous employons aussi la bibliothèque Google Speech API [2] en boîte noire (système propriétaire). Ces deux moteurs de reconnaissance ne sont pas fondés sur le même modèle : les HMM pour PocketSphinx et les réseaux de neurones récurrents (RNN) pour Google. Ces deux systèmes seront donc utilisés de manière complémentaire dans notre architecture.

3.2.3 Fusion de systèmes

En dépit du fait que la reconnaissance vocale fonctionne plutôt bien dans des contextes audios maîtrisés avec un locuteur proche du microphone, la précision chute drastiquement dès que les conditions d'utilisation s'écartent de l'utilisation optimale du système. Ainsi, de nombreux travaux se sont axés sur l'amélioration de la précision des systèmes par fusion, notamment par la combinaison de plusieurs canaux audios.

Par exemple, dans un contexte d'interaction robotique (Onuma et al., 2012), les auteurs ont implémenté un banc de filtres sur les 4 canaux audios du capteur Kinect. La connaissance de

2. http ://googleresearch.blogspot.fr/2015/08/the-neural-networks-behind-google-voice.html

la direction de l'utilisateur permet d'en réduire les interférences. La position de l'utilisateur est estimée à l'aide de l'algorithme de création de squelette du capteur Kinect. Les auteurs se servent ensuite de cette information pour paramétrer le banc de filtres. Ils améliorent ainsi la précision de la reconnaissance de 15%. L'intérêt de ces travaux réside dans la prise en compte de la distance de l'utilisateur pour améliorer la qualité du signal perçu.

De la même manière, (Maganti et al., 2007) utilisent un algorithme de formation de faisceaux pour filtrer les interférences. La connaissance de la position de l'utilisateur à l'aide d'un algorithme de suivi audio visuel permet de diriger le faisceau. Un algorithme d'adaptation utilisant la technique MLLR (Maximul Likelihood Linear Regression) est aussi utilisé afin d'améliorer les modèles acoustiques pour le contexte de l'application. Le MLLR, ou régression linéaire par maximum de vraisemblance, permet de mettre à jour les GMM des modèles acoustiques par une convolution avec une fonction de transformation apprise durant l'estimation. Cela a pour effet d'améliorer la reconnaissance vocale pour le locuteur considéré. Le dispositif est composé d'un cercle de microphones disposés au centre de la salle et d'une caméra RGB externe permettant de visualiser toute la salle. Cette technique entraîne une diminution du taux d'erreur de mot (WER, Word Error Rate) de 10%. Les auteurs rapportent aussi une amélioration des résultats en utilisant la technique de localisation audio visuelle par rapport à la technique de localisation audio seule. Cette technique est à rapprocher de (Seltzer and Stern, 2006), où un algorithme de formation de faisceaux est utilisé pour améliorer la reconnaissance dans les environnements réverbérants.

D'autres travaux ont été centrés sur l'utilisation de plusieurs micros de différents types, répartis de façon aléatoire dans l'environnement. Ainsi, dans (Trawicki et al., 2012), les auteurs présentent une combinaison de différents signaux audios basée sur la quantité d'information présente dans le signal. Ils formalisent ainsi le signal à estimer comme une combinaison linéaire des signaux acquis par chaque micro. Les auteurs obtiennent alors une précision de 94,4% avec une pondération de chaque signal, celle-ci se fondant à la fois sur l'information présente dans un signal et sur la position relative du micro. Cela représente une amélioration de 30% de la qualité du signal par rapport à chaque signal extrait seul. Ces travaux se focalisent sur la seule amélioration du signal, sans présenter de résultats associés à de la reconnaissance vocale.

Enfin, dans les travaux de (Zhang et al., 2015), un seuil est affecté à chaque canal en fonction de leur précision associée. Ils utilisent ainsi un système d'acceptation ou de rejet des descripteurs pour améliorer les performances de reconnaissance vocale. Ils rapportent alors une diminution du WER de 50% en environnement bruité (bruit blanc, bruit d'avion, bruit d'usine) tout en ayant des performances similaires en l'absence de bruit.

Nous pouvons constater que la plupart des systèmes de fusion permettent d'améliorer la robustesse de la chaîne de perception au bruit et à la variation de la distance entre l'utilisateur et les micros. Cependant, la plupart des travaux existants se focalisent sur la fusion de systèmes au niveau bas (signal, descripteurs, etc.). Très peu de travaux fusionnent les informations renvoyées par les systèmes de reconnaissance au niveau symbolique. Ainsi, dans la deuxième partie de ce chapitre (section 3.4), nous présentons un système de fusion bayésienne combinant à la fois les micros et les systèmes de reconnaissance vocale.

3.2.4 Interaction conversationnelle

Toutes les méthodes présentées plus haut peuvent être utilisées comme une information pour un système d'interaction conversationnelle. De nombreux travaux existent sur ce problème d'interaction et notamment dans (Allen et al., 2000), où les auteurs posent la base théorique de ce que doit être l'architecture générale d'un système automatique de dialogue oral. Le système est composé d'une chaîne de perception, construite à partir d'un système de reconnaissance vocale et d'un extracteur d'interprétation. Il contient aussi un gestionnaire d'interaction lié à la gestion du contexte, sans oublier la phase de réponse formée par la synthèse vocale. Ces travaux peuvent donc servir de point de départ pour la création d'un système d'interaction homme-robot.

Le cœur du problème de l'interaction homme-machine se situe en général dans le gestionnaire d'interaction (dialogue manager en anglais). Celui-ci peut parfois se résumer à une machine à états statique, comme dans (Spiliotopoulos et al., 2001) dont le but est la création d'un robot d'assistance médicale. Il s'agit de créer une sorte de base de données parlante, la tâche du robot se résumant à la livraison de médicaments et à la présentation d'informations. Cela justifie l'usage d'une machine à états statique.

Cependant, ce genre de système peut devenir trop restreint lorsque nous cherchons une interaction plus naturelle. Ainsi les travaux présentés dans (Young et al., 2013) font un état de l'art des techniques les plus récentes axées sur une gestion de dialogue par POMDP (Partially Observable Markov Decision Processes). Ces techniques permettent de créer par apprentissage un réseau contenant les états possibles de dialogue ainsi que les probabilités de transition entre ceux-ci. C'est actuellement l'une des techniques permettant de créer les systèmes les plus flexibles. Les POMDP nécessitent cependant des corpus d'interaction pour pouvoir être appris, ce qui, dans notre contexte (recherche d'objets), est difficile à obtenir.

En général, un système de dialogue est centré sur une tâche particulière. Par exemple, dans un contexte proche du nôtre (Kruijff et al., 2006), les auteurs se servent d'un système de dialogue pour aider à la création d'une carte à partir de la perception visuelle de l'environnement. Le robot finit ainsi par créer une carte sémantique contenant la position des portes dans la pièce et autre élément important présent. Ainsi, lorsque le robot détecte une petite ouverture entre deux murs, il demande à l'utilisateur si cela correspond à une porte. L'utilisateur donne alors une réponse permettant de mettre à jour la carte de l'environnement.

Les mêmes genres de travaux ont été menés dans (Lemaignan et al., 2011) où le système robotique se met à jour, cette fois dans un contexte de perception d'objets. Le robot perçoit ainsi un objet inconnu et demande à l'utilisateur ce qu'il représente. Le système de dialogue permet alors de mettre à jour une série d'informations dans une base de donnée hiérarchique formalisée sous forme d'ontologies. Les ontologies permettent de créer une relation entre différents concepts et sont majoritairement utilisées dans le domaine du web sémantique.

Suivant ces travaux, nous nous sommes orientés vers un système de machine à états statique. En effet, dans notre contexte, l'utilisateur doit demander au robot la position d'objets perdus. La tâche reste relativement simple, les difficultés se situant principalement au niveau de la perception. L'utilisateur étant à domicile, le niveau de bruit peut changer radicalement, ce qui aura tendance à dégrader le SNR. Nous proposons une architecture de dialogue classique se fondant sur les travaux

présentés dans (Spiliotopoulos et al., 2001) afin d'apporter des améliorations aux différentes parties de cette architecture générale d'interaction orale.

3.3 Architecture d'interaction

Nous commençons ce chapitre par la description générale d'une architecture d'interaction et présentons ensuite l'implémentation dédiée à notre tâche de recherche d'objets.

3.3.1 Architecture générale de l'interaction

Comme décrit plus tôt, la chaîne d'interaction se compose de 3 parties représentées sur la figure 3.1. La partie bleue représente toute la chaîne gérant la perception de l'utilisateur. Ainsi, le système d'interaction commence par recevoir le signal audio qu'il segmente ensuite à l'aide d'un algorithme de détection d'activité vocale en zones de parole et de non-parole. Ces zones de paroles sont transmises à un algorithme de transcription permettant d'extraire les mots prononcés dans le signal. Un module d'interprétation analyse une première fois la phrase de manière à en extraire les concepts les plus simples pouvant aider le robot à comprendre l'utilisateur.

Les résultats extraits par la chaîne de perception sont ensuite envoyés au module de gestion de l'interaction (orange). Ce module est lié à un historique qui regroupe les changements précédents étant intervenus dans l'environnement, les phrases prononcées et les interprétations associées. Il est aussi lié à une base de données contenant l'état de l'environnement statique (long terme : position des meubles, lieu de l'expérience, etc.), et celui de l'environnement dynamique (court terme : position de l'utilisateur, du robot, des objets déplaçables, etc.). Ce module de gestion de l'interaction peut ainsi formuler une réponse à partir de toutes ces informations pour guider l'utilisateur dans la réalisation de la tâche requise.

Une fois la réponse adéquate conceptualisée par le robot, il la formule à l'aide de la synthèse vocale par ses haut-parleurs. Il peut aussi compléter cette réponse verbale par un comportement non-verbal en affichant des informations complémentaires à l'aide d'un écran ou de LEDs (*Light-Emitting Diode*) embarquées. Il peut encore exprimer des directions de manière non-verbale à l'aide des bras ou de l'orientation de la tête.

3.3.2 Implémentation de l'architecture générale

Suivant ce modèle, nous avons ainsi implémenté notre architecture d'interaction et nous en décrivons les différentes parties dans les sous-sections suivantes.

3.3.3 Signal audio et détection d'activité vocale

La chaîne de perception débute par l'extraction des buffers audios d'un capteur Kinect embarqué sur le robot et d'un smartphone Android posé près de l'utilisateur. Ce dernier permet d'obtenir un signal plus propre lorsque l'utilisateur est loin du robot.

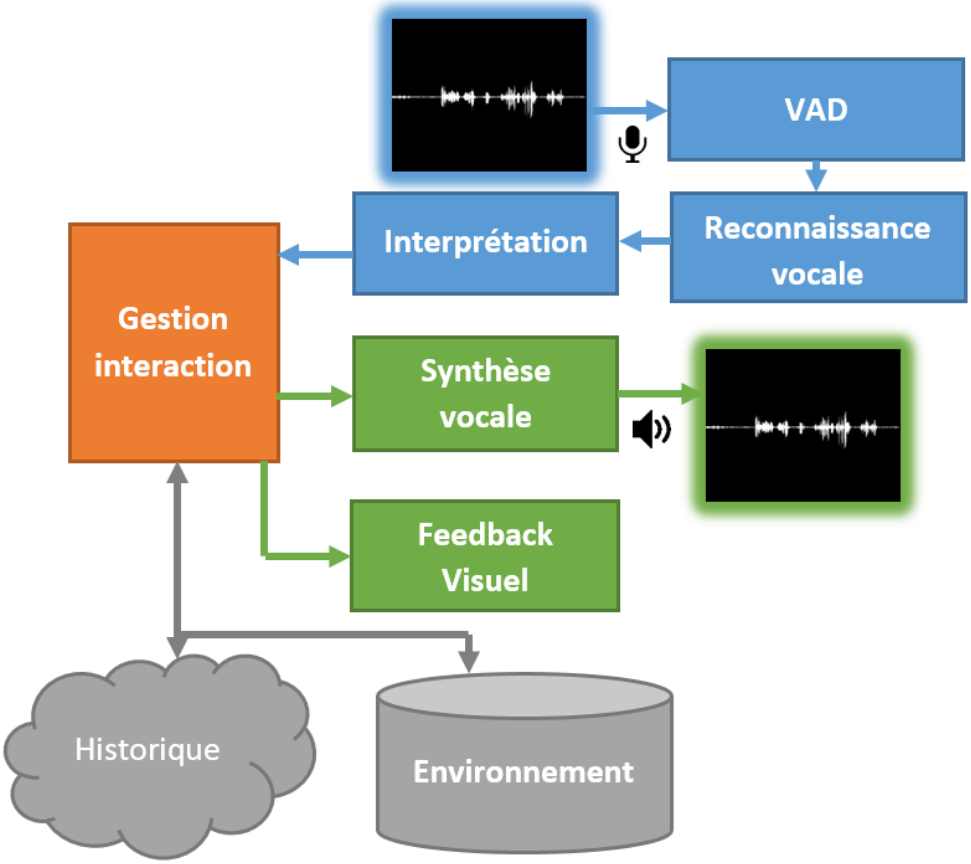


FIGURE 3.1 – Architecture générale du module d'interaction.

Dans notre application, nous avons utilisé le module de détection d'activité vocale de SphinxBase. Cette bibliothèque offre un algorithme de VAD basé sur l'énergie du signal qui est directement compatible avec la bibliothèque de reconnaissance vocale que nous utilisons par la suite. De plus, la plupart des détecteurs OpenSource fonctionnent sur des données hors-ligne, alors que le détecteur de SphinxBase offre la possibilité d'une utilisation en ligne. Ceci est très important pour une application robotique interactive ou l'utilisateur n'attend pas que le robot ait acquis la totalité du signal pour commencer à le traiter.

3.3.4 Transcription

Pour la transcription de la parole, nous avons utilisé le moteur de reconnaissance présent dans PocketSphinx, permettant lui aussi de faire de la reconnaissance vocale en ligne. Par défaut, ce moteur utilise les modèles acoustiques et les modèles de langage du LIUM pour la reconnaissance vocale du français.

Comme la plupart des algorithmes actuels, le moteur de reconnaissance PocketSphinx est basé sur une modélisation par HMM et un décodage des mots à l'aide d'un algorithme de Viterbi. Les

descripteurs utilisés sont les MFCC classiquement employés dans la communauté de la reconnaissance de la parole. Pour extraire ces coefficients, nous calculons la transformée de Fourier discrète (DFT) (équation 3.1) sur un signal de $N = 256$ échantillons x_m, convolué avec une fenêtre de Hamming. Une fenêtre est extraite tous les $M = 100$ échantillons par décalage.

$$y_k = \sum_{n=0}^{N-1} x_n \exp^{\frac{-j2\pi kn}{N}}, \ k \in [\![0, N-1]\!] \tag{3.1}$$

Les fréquences extraites y_k sont ensuite converties dans l'échelle de Mel (équation 3.2) (Bogert et al., 1963), celle-ci permettant d'imiter au mieux le fonctionnement de l'oreille humaine. En transformant ces coefficients z_k par une transformée de Fourier inverse, nous passons dans le domaine cepstral et nous obtenons alors les descripteurs MFCC.

$$z_k = 2595 \log_{10}(\frac{f}{700} + 1) \tag{3.2}$$

Un signal de parole annoté au niveau des phonèmes est ensuite appris à l'aide d'un algorithme de type EM (Expectation Maximization) ou Viterbi pour créer les modèles HMM (Rabiner and Juang, 1993). PocketSphinx permet d'apprendre ses propres modèles acoustiques phonétiques. Cependant, comme nous ne disposons pas de données en quantité suffisante, nous avons utilisé les modèles du LIUM.

Les modèles de langage sont appris sur un corpus de texte. Ici aussi, comme nous ne disposons pas de corpus suffisant, nous avons choisi de créer une grammaire, centrée sur la tâche de recherche d'objets. Cela permet d'obtenir de meilleurs taux de reconnaissance par rapport à une utilisation directe des modèles du LIUM. En effet, le vocabulaire utilisé dans les textes du corpus ne correspond pas du tout à une interaction robotique. Ainsi le système a tendance à former des phrases hors-contexte, ce que nous essayons de résoudre par l'utilisation d'une grammaire plus spécifique à notre tâche.

Nous nous sommes aussi servis du système Google Speech API de Google. Bien qu'il n'y ait que peu d'informations sur celui-ci, il apparaît que Google commence à s'intéresser aux techniques de Deep Neural Network. À l'origine ce système est conçu pour des applications de type « assistant vocal », et est donc ciblé grand vocabulaire. Cette API étant propriétaire, nous n'avons pas accès à ses modèles acoustiques et ses modèles de langage. Nous devons donc l'utiliser telle quelle.

Ainsi, nous avons d'un côté un système Open Source fondé sur PocketSphinx avec une grammaire spécifique, dont les modèles acoustiques ont été appris sur les corpus du LIUM (principalement des journaux d'information radiophoniques). De l'autre, nous avons un système très grand vocabulaire utilisable uniquement en boîte noire, destiné à être employé comme assistant vocal. Ces deux systèmes étant optimisés pour des applications différentes, ils seront utilisés de façon complémentaire dans notre application robotique.

Mesures du taux d'erreur de mots

Le Word Error Rate (WER), ou taux d'erreur de mots, est une mesure généralement utilisée pour caractériser un système de reconnaissance vocale. Cette mesure est définie dans l'équation 3.3

où N est le nombre de mots de référence et S représente le nombre de substitutions (les mots mal reconnus). Par exemple, « cuvette » au lieu de « lunettes » sera une substitution. D est le nombre de suppressions (les mots omis lors de la transcription), et I symbolise le nombre d'insertions (les mots ajoutés durant la reconnaissance). Ce taux d'erreur peut être exprimé comme un pourcentage et il peut être supérieur à 100% s'il y a trop d'insertions.

$$WER = \frac{S + D + I}{N} \tag{3.3}$$

Cette mesure peut aussi être déclinée sous la forme d'une précision de reconnaissance (Word accuracy, WAcc), qui est définie dans l'équation 3.4, H représentant le nombre de mots correctement reconnus. Comme le WER peut être supérieur à 1, le WAcc peut être négatif. Le WER est utilisé dans la suite pour évaluer notre algorithme de fusion de systèmes combinés.

$$WAcc = 1 - WER = \frac{N - S - D - I}{N} = \frac{H - I}{N} \tag{3.4}$$

3.3.5 Interprétation et gestion de l'interaction

Dans notre contexte, le robot connaît ou a construit une carte de l'environnement qui est envisagée comme fixe pendant la durée de l'expérience : cela correspond aux connaissances statiques du robot. De même, le robot connaît la position d'un certain nombre d'objets cibles dans cette même carte qui peuvent être déplacés au cours du scénario. Cette position des objets fait partie des connaissances dynamiques du robot et peut être amenée à évoluer durant les expériences. Le robot doit donc tenir compte de ces informations durant l'interaction.

À l'instar de l'interaction homme-homme, les phrases permettant l'interaction homme-robot sont structurées de différentes manières. Il y a par exemple des phrases interrogatives : « Où se trouve mon parapluie ? », des phrases assertives : « Mes chaussures sont dans l'entrée. », ou bien encore des phrases négatives : « Tu n'es pas un robot. ». Ainsi les phrases interrogatives correspondent souvent à une requête, tandis que les phrases assertives servent de réponse, permettent de valider des informations, ou bien les deux en même temps. Par exemple, la phrase « C'est bon, j'avais bien mis la télécommande sous la table basse. » doit permettre au robot de comprendre que sa réponse a aidé l'utilisateur tout en permettant de valider l'emplacement de l'objet dans l'environnement.

Nous avons implémenté une version primitive de ce module en utilisant le repérage de mots clefs dans la phrase. Cela nous permet d'interpréter cinq types de phrases : la recherche d'objets, la confirmation, l'infirmation, la salutation et la fin de conversation.

Le système de gestion de l'interaction est basé sur une machine à états statique schématisée dans la figure 3.2. Le robot commence par demander à l'utilisateur de confirmer qu'il a bien besoin d'aide. Après une réponse positive de sa part, le robot lui demande ce qu'il veut faire. La tâche étant la recherche d'objets, l'utilisateur demande au robot de trouver un objet perdu dans l'environnement. Le robot lui répond alors en indiquant le meuble sur lequel l'objet est posé. L'utilisateur pourra ensuite demander d'en trouver d'autres ou pourra mettre un terme à l'interaction.

La gestion d'erreur s'effectue à l'étape de recherche d'objets. Si le robot a interprété une phrase comme étant une action de recherche, mais qu'il ne connaît pas l'objet mentionné, il demandera à

l'utilisateur de poser une nouvelle question en précisant qu'il n'a pas reconnu l'objet.

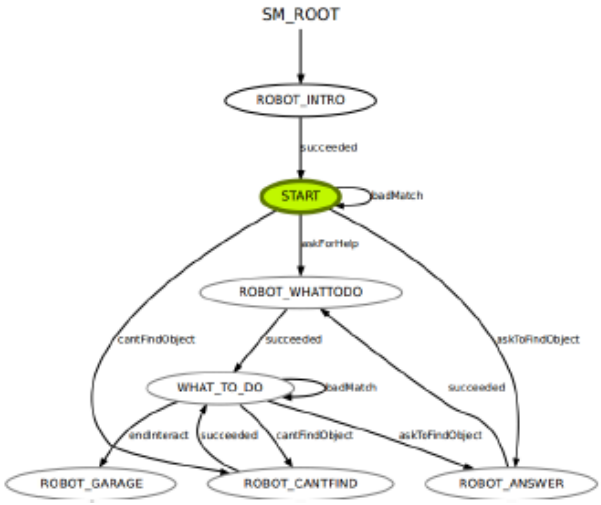


FIGURE 3.2 – Machine à états pour l'interaction visualisée sous Smach.

Ce système d'interaction remplit les fonctions minimales requises pour l'interprétation des phrases et la tâche de recherche d'objets et laisse donc beaucoup de place à l'amélioration. L'interprétation pourrait notamment être grandement améliorée par un module d'analyse morpho-syntaxique. Nous pourrions alors créer une grammaire qui prenne en compte aussi bien les mots clefs (comme c'est déjà le cas) mais aussi les fonctions grammaticales pour augmenter la couverture lexicale de l'interprétation, voire l'apprentissage de nouveaux types de phrases. De plus, pour le moment l'interaction est mono-tâche : il s'agit d'une recherche d'objets, sans possibilité de demander ou apporter des informations complémentaires. Par exemple :
usager : « Je cherche la tasse. »
robot : « Je l'ai trouvé sur la table. »
usager : « Non, pas la verte, la bleue. »
Cela laisse sous-entendre qu'une nouvelle tasse existe et que le robot ne connaît pas encore son emplacement. Il doit ainsi mettre à jour son environnement dynamique.

3.3.6 Synthèse vocale

La synthèse vocale est primordiale en interaction homme-robot. En effet, bien que certains robots soient équipés d'un écran, l'interaction semble plus naturelle si elle est conduite oralement. Pour cela, nous avons utilisé le système de synthèse de Google Traduction qui consiste à envoyer l'adresse URL contenant la phrase à prononcer. Un fichier Wave contenant le signal audio à prononcer est ensuite téléchargé et peut être joué directement par un module de lecture audio embarqué sur le robot.

3.3.7 Discussion

Grâce à ces différents développement, nous avons pu concevoir une chaîne d'interaction primitive remplissant les objectifs fixés pour la tâche de recherche des objets. Cependant, elle est peu robuste aux changements importants du contexte sonore, et notamment à la variation du SNR. Ce ratio peut changer du fait des variations du bruit ambiant, ou si l'utilisateur devient trop distant par rapport au robot. Dans les prochaines sections, nous nous attachons à l'amélioration de ces différentes parties de l'architecture d'interaction.

3.4 Amélioration de la perception : fusion bayésienne de moteurs de reconnaissance

Tout d'abord, nous nous sommes intéressés à la chaîne de perception (microphone, VAD, et moteur de reconnaissance vocale).

Dans le contexte de la robotique d'assistance à domicile, le bruit peut être omniprésent dû à l'environnement (bruits de fonds, circulation, TV, etc.), ou généré par les moteurs du robot. Dans cette situation, il faudrait donc entraîner (ou du moins adapter) les modèles des systèmes de reconnaissance vocale pour diminuer le WER. Cependant, nous ne disposons pas de base de données assez conséquente pour cela. De plus, lors de chaque session d'interaction, l'utilisateur n'est jamais exactement à la même place. Cela peut être dû à la présence d'un obstacle empêchant le robot de se positionner correctement, ou bien au fait que l'utilisateur s'est déplacé durant la conversation. Le niveau de bruit peut aussi varier du fait des activités de l'usager (ouverture d'une fenêtre, augmentation du volume de la TV, etc.), rendant l'environnement sonore très variable.

Considérant ces paramètres, nous avons centré notre travail sur l'utilisation des systèmes de reconnaissance existants bien qu'ils ne soient pas bien adaptés à notre contexte. Nous avons choisi d'utiliser plusieurs systèmes de reconnaissance vocale conçus pour fonctionner dans différents contextes, ainsi que différentes entrées sonores pour avoir plusieurs sources avec différents SNR.

Pour répondre à ces problèmes perceptuels et pouvoir exploiter au mieux tous les systèmes et capteurs, nous proposons une approche qui consiste à combiner M flux audio, K systèmes de détection d'activité vocale et N moteurs de reconnaissance vocale. Dans cette section, nous proposons une formalisation bayésienne du problème de fusion de $C = M * K * N$ systèmes globaux connaissant *a priori* la distance du locuteur dans le but d'améliorer le WER global.

Cette distance peut par exemple être extraite à l'aide d'un capteur Kinect calibré, d'autant plus que celui-ci est déjà utilisé dans la détection d'intentionnalité.

3.4.1 Formalisme et architecture proposés

Pour ces travaux, nous nous sommes inspirés des travaux de (Herranz et al., 2015) qui présentent un système de fusion bayésienne d'informations dans le domaine de la vision par ordinateur. Il fusionne ainsi un détecteur de type de plat et la position géolocalisée de l'utilisateur pour en déduire le type de plat pris en photo par le smartphone. Les auteurs modélisent ces informations à l'aide d'un réseau bayésien. La géolocalisation donne accès au restaurant probable où se trouve l'utilisateur,

et aux recettes disponibles dans celui-ci pour pouvoir comparer les informations avec la sortie du détecteur. Nous proposons un système de fusion similaire permettant d'exploiter les dépendances entre la distance du locuteur, les hypothèses de reconnaissance générées et le système à choisir. Le système est généralisé à M microphones, K dispositifs de détection d'activité vocale et N moteurs de reconnaissance vocale. Nous avons ainsi potentiellement $C = M * K * N$ systèmes combinés représentés sur la figure 3.3.

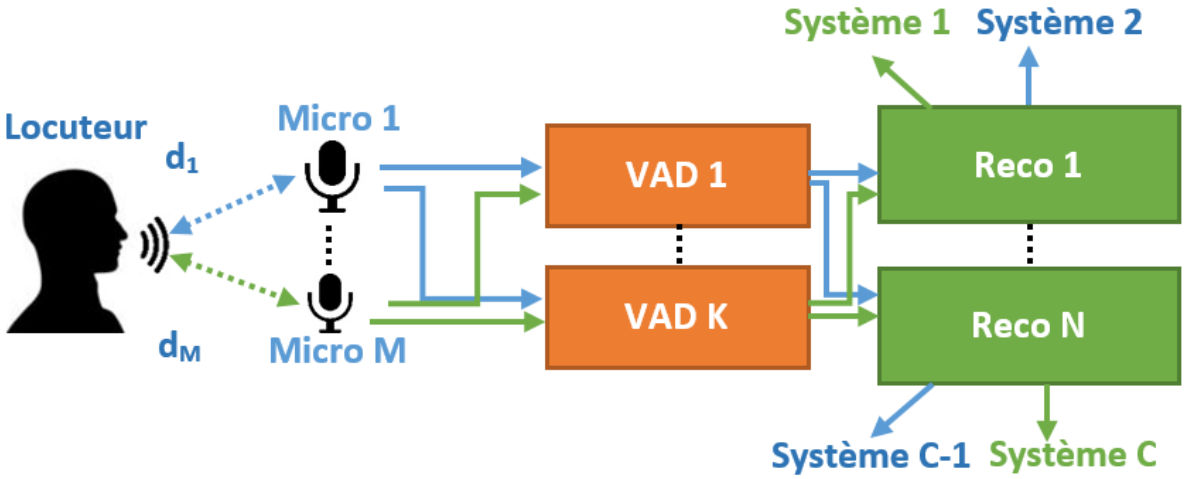


FIGURE 3.3 – Combinaisons engendrées par le système comportant plusieurs micros, algorithmes de VAD et moteurs de reconnaissance vocale.

Le but, pour chaque phrase prononcée, est de sélectionner de la meilleure combinaison (ou système combiné), connaissant les distances séparant le locuteur des différents microphones. Tous les algorithmes sont lancés en même temps sur le même énoncé, ce qui donne dans le meilleur des cas, une phrase reconnue par système combiné. Cela peut se formaliser par le graphique bayésien illustré sur la figure 3.4 où d représente la distance séparant le locuteur de chaque microphone. U est l'ensemble des hypothèses pour chaque système, et S symbolise le système choisi. Enfin, U_S représente l'hypothèse émise par le système S.

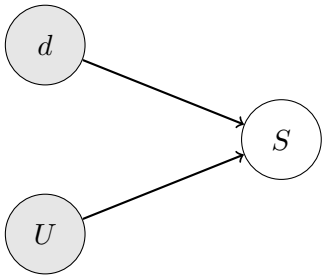


FIGURE 3.4 – Modèle graphique probabiliste utilisé pour choisir le bon système.

d et U sont des variables aléatoires observées lorsqu'une phrase est prononcée. À partir de celles-

ci, il faut déterminer l'équation 3.5 qui est la probabilité de reconnaître une phrase correctement avec le système S compte tenu de la distance d et l'ensemble des phrases renvoyées $U = \bigcup_S$ par chaque système combiné :

$$P(S|d,U) \propto P(S|d)[U_S \neq \emptyset] \tag{3.5}$$

$P(S|d)$ est la probabilité que le système global S soit le plus précis, connaissant la distance entre le locuteur et le micro. Cette probabilité est inversement proportionnelle au taux d'erreur de mot de chaque système en fonction de la distance. Enfin, $[U_S \neq \emptyset]$ est égal à 1 si une phrase U_S est renvoyée par le système S. S'il n'y a pas de phrase reconnue, cette probabilité devient nulle. Le but est donc d'apprendre les densités décrivant $P(S|d)$, pour ensuite estimer le résultat de l'équation (3.6) connaissant la distance de l'utilisateur :

$$\underset{S}{argmax}(P(S|d,U)) \propto \underset{S}{argmax}(P(S|d)[U_S \neq \emptyset]) \tag{3.6}$$

3.4.2 Implémentation et intérêt dans notre contexte

Dans notre contexte robotique, nous avons utilisé deux microphones : le micro présent sur le smartphone Android, et un des micros du capteur Kinect. Concernant la reconnaissance vocale, nous avons employé les deux moteurs présentés dans les sections précédentes. En se servant du détecteur d'activité vocale de SphinxBase, les quatre combinaisons illustrées figure 3.5 sont ainsi obtenues.

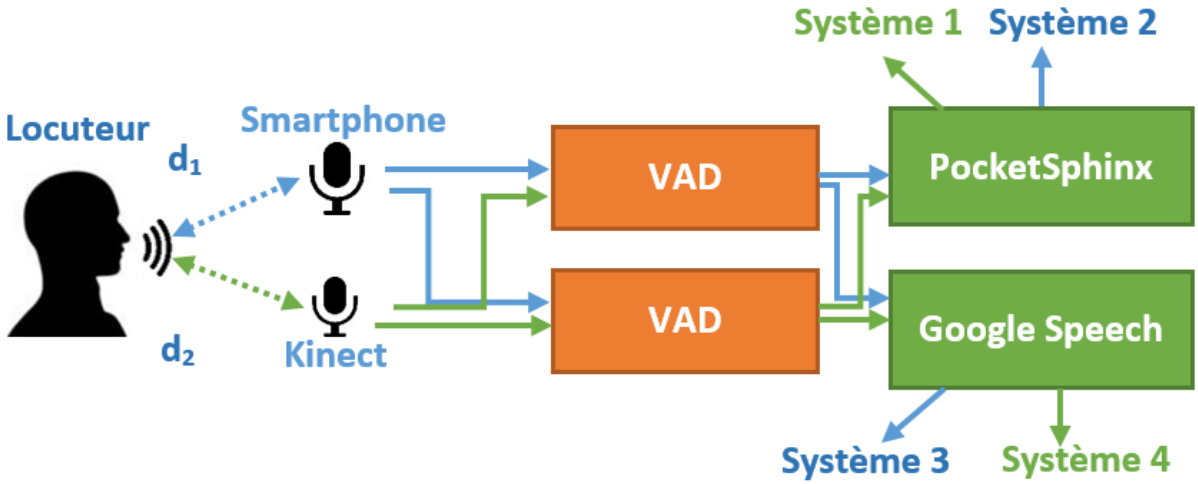


FIGURE 3.5 – Combinaisons engendrée par le système multi-canal et regroupant plusieurs API de reconnaissance vocales.

Ces combinaisons peuvent être justifiées par les caractéristiques de chaque composante des systèmes combinés :

Microphones : ils ont en général des sensibilités différentes, ce qui a plusieurs effets. Lorsque la sensibilité est trop forte, le microphone a tendance à saturer dès que le locuteur se rapproche. Lorsqu'elle est trop faible, le micro fonctionne très mal dès que la distance avec l'utilisateur augmente. Dans les deux cas, cela va se traduire par une baisse de la qualité de la reconnaissance pour au moins deux raisons. D'une part, le détecteur d'activité vocale ne va pas se déclencher correctement en particulier pour les micros ayant une sensibilité trop basse ou si la parole est trop distante. D'autre part, les moteurs de reconnaissance sont entraînés pour certaines distances entre les micros et les locuteurs (et donc pour un certain SNR). Pour les émissions de radio ou l'assistance vocale, les locuteurs sont en général plutôt proches des microphones et le SNR est donc élevé.

Dans notre application, nous utilisons d'une part le micro d'un smartphone Android qui est *a priori* calibré pour une prise de son à très faible distance (appels téléphoniques, assistant vocal, etc.) : le micro a donc une faible sensibilité. Et d'autre part, nous utilisons le microphone du capteur Kinect, qui est conçu pour fonctionner à plus grande distance (environ 2m) pour des applications telles que les jeux vidéos ou bien l'assistance vocale pour la console X-Box. Nous avons donc deux types de micros qui ont *a priori* un fonctionnement optimal sur des plages de distance différentes.

Moteurs de reconnaissance : La différence entre les moteurs se situe à la fois au niveau de la technique employée (HMM, RNN), mais aussi au niveau des bases de données utilisées pour la créations des modèles acoustiques et de langages, des grammaires, ou bien encore des dictionnaires utilisés.

Pour la bibliothèque PocketSphinx, les modèles acoustiques que nous avons utilisés sont ceux du LIUM créés durant la campagne d'évaluation ESTER de corpus radiophonique. Les modèles de langages sont eux aussi entraînés à partir du même corpus, et donc complètement hors de notre contexte applicatif. Ainsi dans notre contexte, le mot « chaise » aura une forte probabilité d'apparition, alors que dans les modèles du LIUM, le mot « chine » sera beaucoup plus présent tout en étant phonétiquement proche. Cela risque de générer des erreurs de reconnaissance. Pour disposer d'un système dédié à la tâche visée, nous avons donc développé une grammaire qui couvre les requêtes et les réponses possibles de l'utilisateur. Cela a pour effet de restreindre la couverture lexicale et le nombre de phrases reconnaissables de notre système. A l'opposé, nous avons aussi utilisé l'API Google Speech. Cet outil propriétaire étant créé par Google, il n'y a que très peu d'informations sur son fonctionnement, même si c'est *a priori* un système très grand vocabulaire puisqu'il est utilisé dans l'assistant vocal Google Now©.

Finalement ces deux microphones et outils de reconnaissance vocale se combinent en quatre systèmes qui ont *a priori* des performances différentes suivant le contexte audio (bruit) et la position du locuteur par rapport aux micros.

Caractérisation des systèmes connaissant la distance

Pour pouvoir choisir le meilleur système à chaque instant, il faut commencer par caractériser leur probabilité conditionnelle $P(S|d)$. Pour cela, nous avons créé un corpus d'énoncés en plaçant nos deux microphones au même endroit, et en enregistrant 17 phrases prononcées trois fois chacune à quatre distances différentes du dispositif. Nous obtenons ainsi le corpus résumé dans la table 3.1 composé de 816 enregistrements par micro.

TABLE 3.1 – Corpus utilisé pour la caractérisation et l'évaluation des systèmes combinés. Les enregistrements ont été réalisés pour deux microphones.

Locuteurs	**Phrases différentes**	**Distances**	**Total**	**Durée totale par micro**
4	17	4	816	34'03"

Considérant ce corpus, nous avons calculé le WER par distance d pour chaque système combiné S. Sachant que chaque moteur de reconnaissance renvoie les $NBest$ meilleures hypothèses pour chaque phrase prononcée. Nous utilisons l'hypothèse offrant le WER le plus bas. Nous caractérisons ainsi le fonctionnement optimal de chaque moteur pour les quatre distances. Lorsque la détection d'activité vocale ne s'est pas déclenchée ou lorsque le moteur ne reconnaît rien, le WER est estimé comme nul. Nous avons nommé cette mesure le $U-WER$. Ceci permet de caractériser uniquement les moteurs de reconnaissance. Enfin, pour pouvoir interpoler le comportement des moteurs sur n'importe quelle distance, nous utilisons une régression polynomiale de degré 3 permettant d'avoir le comportement de chaque système en fonction de la distance.

Nous obtenons ainsi les courbes présentées sur la figure 3.6. Une première analyse de ces résultats permet d'observer que la combinaison Kinect + Google Speech donne les meilleurs résultats en interaction proche, tandis qu'elle génère les résultats les moins concluants pour deux mètres de distance. Cependant la combinaison Smartphone + PocketSphinx fonctionne exactement à l'opposé, en donnant de bons résultats pour la parole distante, tout en ayant des performances limitées en interaction proche.

Les courbes sont inversement proportionnelles aux densités représentant $P(S|d)$ que nous pouvons ensuite normaliser. Notre système de fusion est résumé dans l'algorithme 6.

Algorithme 6 : Algorithme de fusion

Résultat : $\underset{S}{argmax}(P(S|d,U))$=Fusion($d$, U, $P(S|d)$)
$P_{max} = 0$
for *pour chaque système* S **do**
    Calculer $[U_S \neq \emptyset]$
    Calculer $P(S|d,U) \propto P(S|d)[U_S \neq \emptyset]$
    if $P(S|d,U) > P_{max}$ **then**
        $P_{max} = P(S|d,U)$
Calculer $S^* = \underset{S}{argmax}(P(S|d,U))$
return S^*

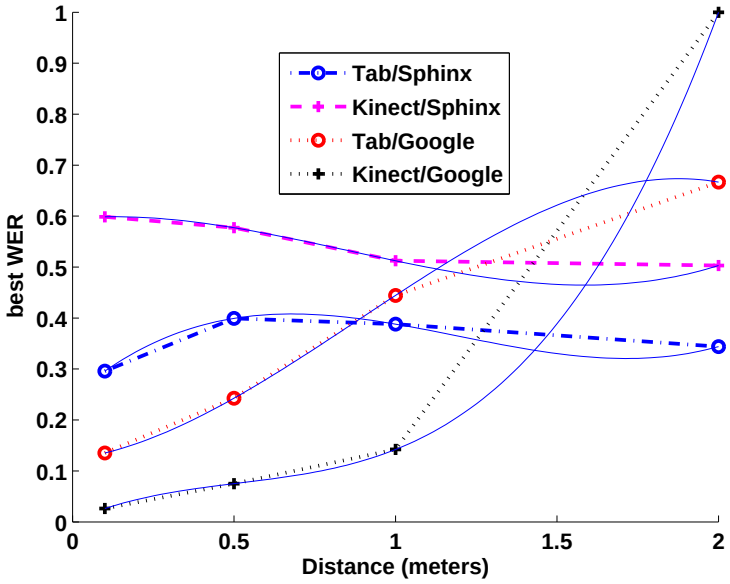


FIGURE 3.6 – Régression polynomiale permettant d'interpoler le WER en fonction de la distance pour les 4 combinaisons testées.

3.4.3 Évaluations

Nous présentons ici les résultats des évaluations de notre système de fusion suivant deux processus différents : la validation croisée et une validation en contexte réel.

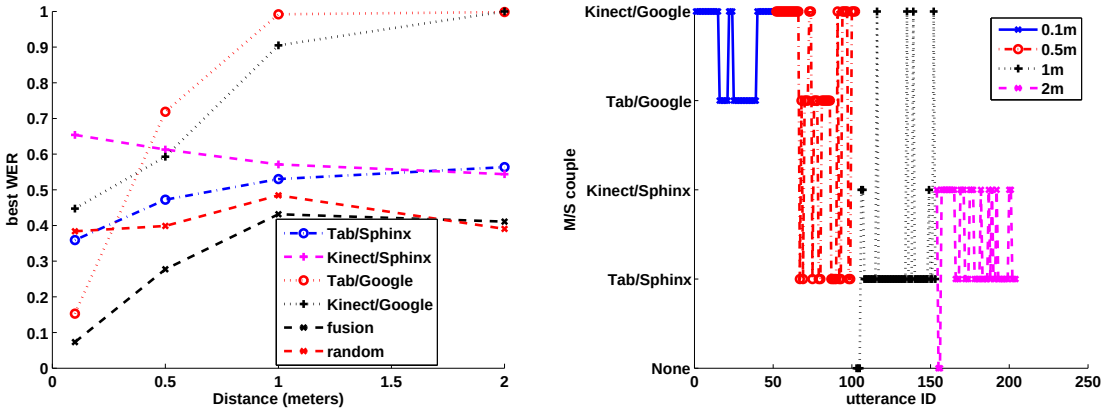
Validation croisée

Dans un premier temps, nous avons utilisé une technique de validation croisée de type « leave-one-out ». En nous servant du corpus présenté dans la table 3.1, nous avons ainsi estimé le U-WER en fonction de la distance pour chaque système combiné S sur trois des quatre locuteurs et avons testé notre système sur le quatrième. Nous avons répété cette procédure pour chaque locuteur. Les résultats obtenus en fonction de la distance sont présentés sur la figure 3.7a en calculant le WER moyen classique. Le WER est égal à 1 lorsqu'aucune phrase n'est reconnue par le système, que le détecteur d'activité vocale se soit déclenché ou pas (contrairement au U-WER, qui est nul dans ce cas).

Nous remarquons que notre système de fusion est plus performant comparé à chaque système combiné S pris séparément. Pour démontrer l'utilité de la phase d'apprentissage des densités $P(S|d)$, nous affichons aussi les résultats pour des densités initialisées aléatoirement (courbe « random »). Nous pouvons constater que notre système est là encore plus performant.

Enfin, sur la figure 3.7b, nous exposons les choix que notre système de fusion a effectués pour chaque énoncé traité durant la phase de test. En abscisse, l'indice de la phrase prononcée est

représenté, et en ordonnée le système sélectionné comme ayant la probabilité de donner la meilleure précision par le biais de la fusion.



(a) Résultats de la fusion

(b) Sélections opérées par la fusion

FIGURE 3.7 – Résultats et comportement de notre algorithme de fusion en fonction de la distance sur le corpus de test (leave one out).

Les résultats sont exposés dans la table 3.2 avec une réduction du WER moyen de 33,3% pour notre système de fusion. L'initialisation aléatoire des densités $P(S|d)$ engendre une amélioration de 6,7%. Cela est dû au fait que nous utilisons tout de même la probabilité $P(S|d,U)$. De meilleures performances du système de fusion sont obtenues par rapport au système capteur Kinect + Google Speech API employé seul avec un gain relatif de 43,8% en WER. Nous passons ainsi de 48,1% de WER en utilisant le meilleur système (combinaison téléphone + PocketSphinx) à 29,8% en utilisant notre système de fusion confrontant les quatre systèmes combinés. Ces résultats de validation croisée donnent des résultats très encourageants pour une expérimentation en condition réelle.

TABLE 3.2 – Résultats en WER de la validation croisée.

Système	**Smartphone/Sphinx**	**Smartphone/Google**	**Kinect/Sphinx**
Erreur (WER)	48,1%	71,6%	59,5%
Système	**Kinect/Google**	**Aléatoire**	**Fusion**
Erreur (WER)	73,6%	41,4%	29,8%

Validation par capture de mouvements

Pour cette deuxième séquence d'expérimentations, nous apprenons les densités $P(S|d)$ sur le corpus de phrases utilisé dans la validation croisée. Pour l'évaluation, nous utilisons un dispositif de capture de mouvements permettant de mesurer en temps réel la distance du locuteur relativement aux micros ; celui du smartphone et celui du capteur Kinect. Le dispositif employé lors de ces

expériences a été conçu par Optitrack [3]. Celui-ci est plus récent que le système de capture de mouvements utilisé pour le filtrage dans le chapitre 2, nous n'avons donc plus besoin de le calibrer avant utilisation. De plus, toutes les expériences étant effectuées dans le repère monde, nous n'avons pas besoin d'effectuer de changements de repère pour le calcul des distances.

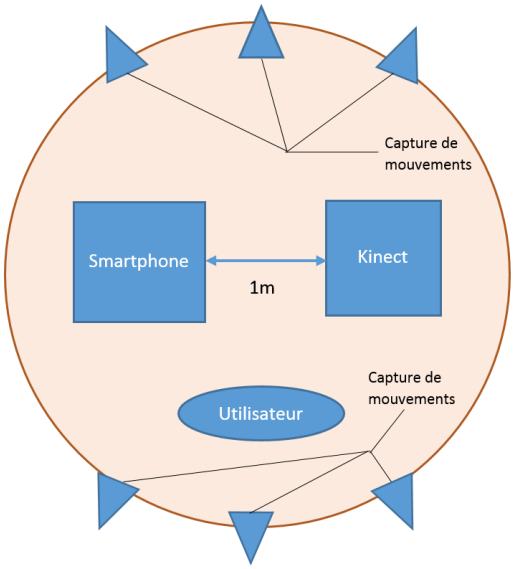


FIGURE 3.8 – Schéma de principe du dispositif de fusion multimodale. L'utilisateur peut se déplacer et parler dans toute la zone orange.

Durant les expériences (voir la figure 3.8, les deux micros sont séparés d'une distance d'environ un mètre. L'utilisateur prononce une liste de 102 phrases en se déplaçant aléatoirement (zone orange) autour et entre les micros (carrés bleus) avant chaque énoncé. Le dispositif de capture de mouvements (triangles bleus) est utilisé pour matérialiser la position de chaque micro. L'utilisateur porte aussi un casque et un gant comportant des marqueurs. Ainsi, à chaque phrase prononcée, le locuteur lève la main pour pouvoir synchroniser les canaux audios et la capture de mouvement. Pour chaque énoncé, nous pouvons donc mesurer les distances entre le casque et les micros pour ensuite les utiliser dans notre dispositif de fusion bayésienne.

Les résultats de ces expériences sont résumés dans la table 3.3.

Nous observons qu'en contexte réel, notre dispositif de fusion reste meilleur avec un WER de 28,4% au lieu de 33,2% pour le système le plus précis durant les expériences. En moyenne, nous obtenons un gain de 26,6% par rapport à chaque système pris seul, ce qui reste cohérent avec les expériences effectuées en validation-croisée. Le faible gain par rapport au meilleur système peut être expliqué par le fait que très peu de phrases ont été prononcées à moins de 1m d'un des microphones, ce qui a pu créer un biais dans le comportement de notre système de fusion.

3. http ://www.optitrack.com/

TABLE 3.3 – Résultats des expériences exprimés en terme « moyenne(variance) » en pourcentage de WER.

Système	**Smartphone/Sphinx**	**Smartphone/Google**	**Kinect/Sphinx**
Erreur	49,3(16,1)%	57.4(20,0)%	33.2(13,3)%
Système	**Kinect/Google**	**Fusion**	
Erreur	79,9(14,0)%	**28,4(11,1)%**	

3.4.4 Discussion

Le dispositif de fusion que nous avons proposé a permis de créer un système robuste aux variations de positions du locuteur durant une session d'interaction. Avec un entraînement minimal du système, nous pouvons ainsi améliorer le WER d'environ 20% avec seulement 2 microphones et 2 API de reconnaissance vocale. Nous pouvons donc imaginer obtenir de meilleurs résultats en augmentant le nombre de combinaisons et par l'ajout de plusieurs détecteurs d'activité vocale.

L'avantage de ce dispositif est qu'il ne nécessite pas de modifications ou de connaissances sur les différentes parties du système. D'autres améliorations pourraient être envisagées en ajoutant des étapes de filtrage entre l'acquisition audio et le détecteur d'activité vocale.

Ces travaux ont donné lieu à des soumissions dans la conférence IEEE-ICASSP 2016 et dans la revue CVIU 2016 (en cours de révision).

3.5 Amélioration de la réponse : feedback visuel

Parfois, la synthèse vocale peut ne pas suffire. En particulier dans les environnements bruités, ou bien pour des personnes souffrant d'une diminution de l'audition. C'est en particulier le cas dans notre contexte, puisque ce trouble touche régulièrement les personnes âgées.

Pour améliorer la compréhension de la réponse formulée, nous nous sommes orientés vers plusieurs solutions visuelles pour compléter la synthèse vocale. Ainsi, la première étape pour la tâche de recherche d'objets a été d'indiquer la position de l'objet perdu par un geste du bras ou de la tête du robot tout en donnant sa position oralement. Des observations lors de campagnes présentées dans le dernier chapitre ont eu tendance à confirmer une amélioration de la qualité de l'interaction.

Toujours dans le but de clarifier l'interaction, nous nous sommes ensuite focalisés sur la création d'un dispositif de feedback visuel à base de signaux lumineux. Ce besoin peut s'expliquer par le fait que la plupart des robots ont des temps de latence, en particulier entre les différents tours de parole. Cela peut générer une certaine confusion pour l'utilisateur qui ne sait alors plus quand parler. Certains robots comme Nao et Roméo intègrent des LEDs permettant de créer des signaux lumineux pour aider l'utilisateur à comprendre quand il doit s'exprimer. Nous avons ainsi conçu une solution similaire à embarquer sur le robot PR2, qui ne possède pas ce type de dispositif.

Un système externe est cependant difficile à intégrer sur le robot. En effet une fois conçus, les robots ne possèdent pas forcément d'accès pour des capteurs ou éléments électriques supplémentaires. Nous nous sommes orientés vers une solution portable pouvant être simplement posée sur ceux-ci et

ne nécessitant pas de modification matérielle. Cette solution présente l'avantage d'être réutilisable par d'autres robots et applicable à d'autres scénarii.

3.5.1 Mise en œuvre

Nous avons choisi un Raspberry Pi 2 équipé d'une clef Wi-Fi et une batterie pour obtenir le système portable, léger, et sans fils. Pour l'affichage nous avons utilisé un écran RGB composé de 32X32 LEDs. Le schéma de principe et une photo du dispositif final sont représentés sur la figure 3.9.

Nous avons décidé de représenter 3 états du robot. Ainsi, l'état présenté en figure 3.10a sera utilisé pour indiquer à l'utilisateur que le robot est prêt à l'écouter. L'état de la figure 3.10b, pour indiquer que le robot est en train de formuler une réponse, et le dernier état sur la figure 3.10c pour indiquer que le robot est dans un état occupé (hors de l'interaction).

Ce dispositif va être validé par une nouvelle campagne d'expérimentations à la fin de projet. Il peut présenter une solution très intéressante pour l'interaction avec les personnes âgées, et les utilisateurs non-experts en général.

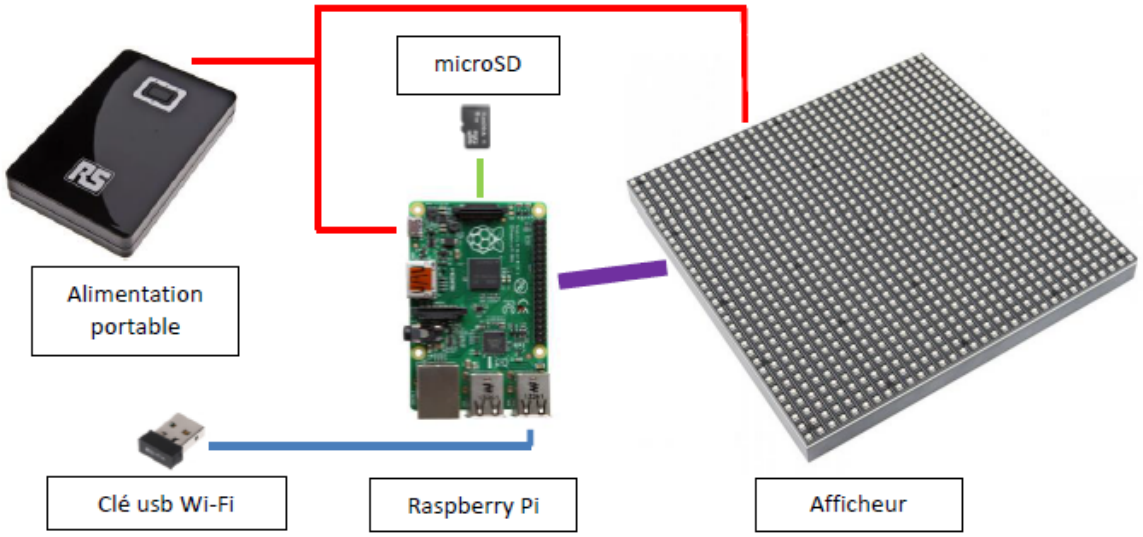
Un autre intérêt de ce dispositif est qu'il peut très facilement évoluer. En effet, un nouvel état peut être créé en ajoutant n'importe quelle image de 32 par 32 pixels, une séquence d'images encodée en gif permettant de lancer une animation. Cela le rend suffisamment générique pour être utilisé à nouveau dans d'autres applications.

3.6 Ontologies : vers une interprétation plus fine du contexte

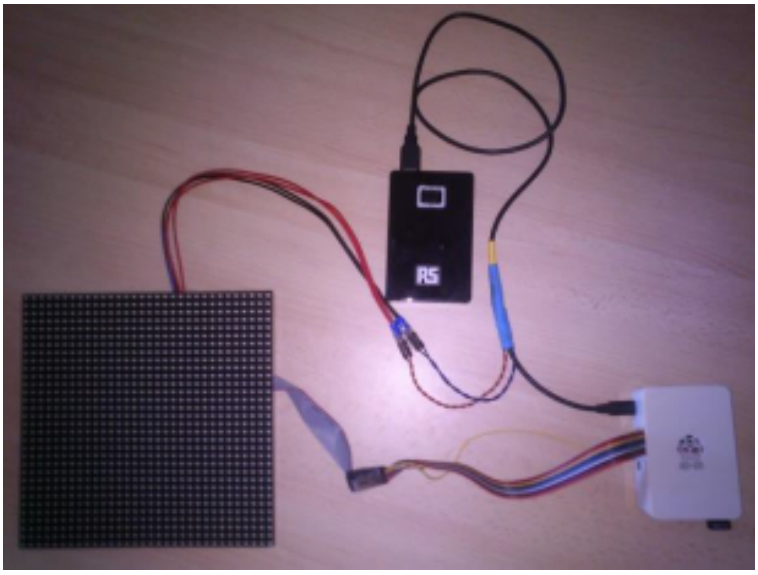
La dernière amélioration à apporter pour la prise en compte du contexte se situe au niveau de l'interprétation et de la gestion de l'interaction. Pour cela, il faut déjà avoir un moyen de représenter au mieux les données décrivant l'environnement. Dans cette section, nous présentons nos travaux préliminaires pour une prise en compte de l'interprétation plus fine de l'environnement.

Les connaissances liées à l'environnement peuvent souvent être représentées de manière hiérarchique. Par exemple, une chaise va se situer dans une pièce et positionnée d'une certaine manière. Celle-ci est aussi un meuble. Nous aurons ainsi la relation entre les concepts suivante : la chaise **est** un meuble. La position **de** la chaise **qui est** un meuble se trouve **dans** la pièce.

Une base de données classique construite à partir d'un modèle de données relationnel ne permettra pas forcément de représenter toutes ces relations. Dans (Lemaignan et al., 2010), les auteurs présentent un serveur d'ontologie permettant de représenter les connaissances du robot de façon hiérarchique tout en mettant à jour les nouvelles informations apportées par l'utilisateur. Dans ces travaux, les informations sont apportées de manière explicite et le robot n'apprend pas de nouveaux concepts d'une formulation implicite. Une ontologie est une structure hiérarchique dont chaque nœud représente un concept et chaque relation peut être décrite par le mot **est**. Ainsi, une représentation par ontologie du concept « chaise » peut être : une chaise **est** un meuble qui **est** un objet qui **est** un gros objet qui **est** une chose. Le concept de chose est la racine (base) de l'ontologie. Nous pouvons aussi créer des instances (ou entités) de ces concepts. Dans notre contexte une instance sera la



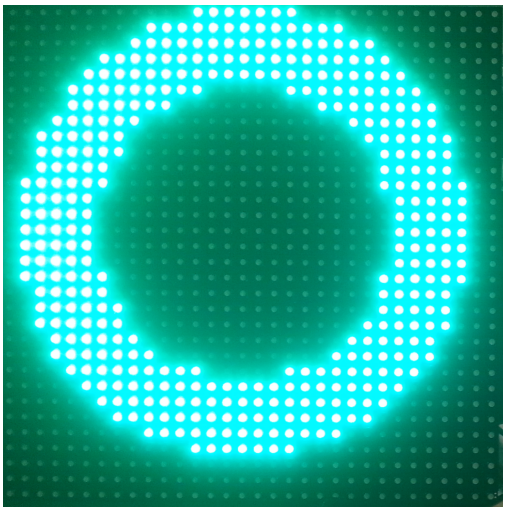
(a) Schéma de principe



(b) Montage

FIGURE 3.9 – Dispositif de feedback visuel.

matérialisation physique du concept de chaise (la chaise bleue, la chaise rouge, la chaise du salon, etc.). Contrairement aux bases de données classiques, les ontologies se basent sur l'axiome de monde ouvert. Ainsi, si une entrée n'existe pas dans la base de données, elle est considérée comme fausse. Tandis que si une entrée n'est pas présente dans l'ontologie, elle est considérée comme inconnue. Par exemple :



(a) Indicateur « écoute »

(b) Indicateur « réponse en cours »

(c) Indicateur « occupé »

FIGURE 3.10 – États du système de feedback visuel

« *La chaise est un meuble.* »
« *Est-ce que l'armoire est un meuble ?* »
Le résultat renvoyé par l'ontologie serait « *Je ne sais pas.* », tandis que le résultat renvoyé par la base de données serait « *Non.* ». Ainsi, l'avantage de ce postulat de monde ouvert dans notre contexte est qu'il prend directement en compte le fait que le robot n'est pas omniscient et que des informations peuvent lui manquer. L'utilisateur pourra alors lui apporter ces compléments.

La possibilité d'adjoindre un algorithme permettant d'inférer des informations à l'ontologie présente un autre avantage pour l'amélioration des connaissances du robot :

« La chaise bleue est une chaise. »
« Toutes les chaises sont des meubles. »
Inférence : « La chaise bleue est un meuble ».

Ainsi, nous pourrons utiliser ce système d'inférence pour essayer de compléter les connaissances du robot, et vérifier que ces connaissances sont cohérentes. Par example, le système donnerait une erreur dans le cas :
« La chaise est un meuble. »
« La chaise est un petit objet. »
« Les meubles sont de gros objets. »
Erreur : La chaise ne peut pas être à la fois un petit et un gros objet.

Nos travaux se sont pour l'instant concentrés sur l'élaboration d'une ontologie liée à notre tâche de recherche d'objets en représentant les relations entre les différents concepts présents dans l'environnement, ce qui est une tâche très délicate. En effet, tout en concevant l'ontologie, il faut garder à l'esprit la tâche ciblée et vérifier que tous les concepts restent cohérents. Pour sa création, nous avons utilisé l'outil de création Protegee [4], exposé sur la figure 3.11.

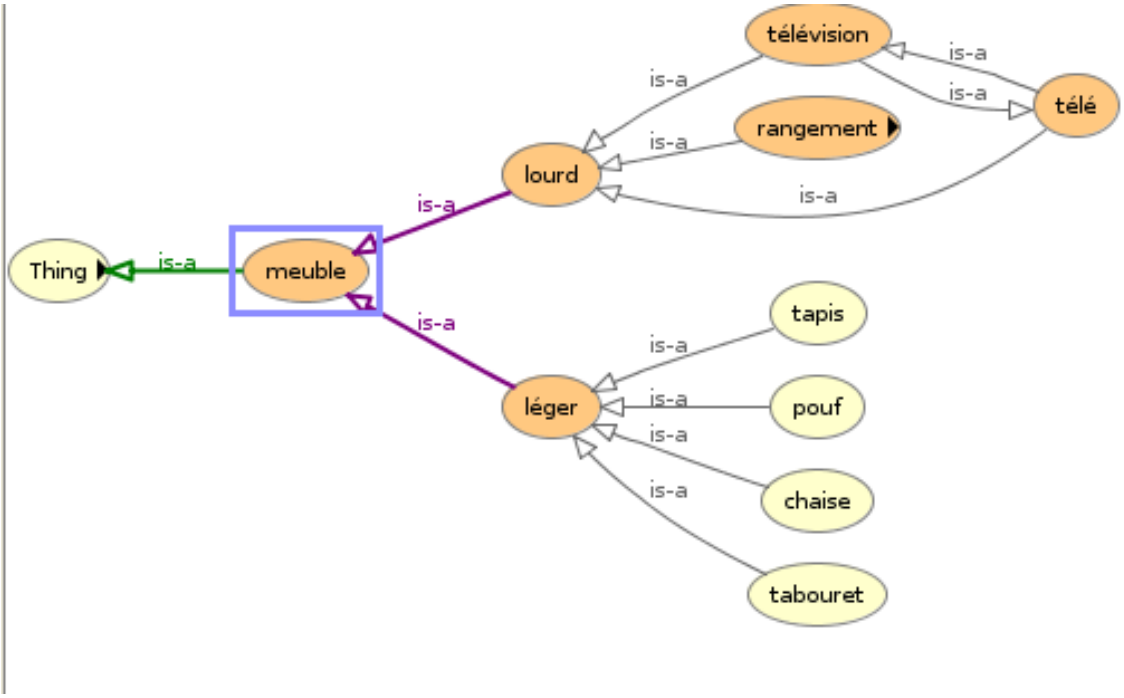


FIGURE 3.11 – Représentation de l'ontologie centrée sur la représentation de l'environnement du robot dans l'outil Protegee.

Cependant un problème de taille se pose et nous l'avons, pour l'instant, peu abordé : la mise à jour des connaissances. En effet, la plupart des ontologies sont utilisées pour le web sémantique (Li

4. http ://protege.stanford.edu/

et al., 2015), (Wu and Etzkorn, 2015), et peu d'applications existent en robotique. La mise à jour est donc problématique dû aux erreurs potentielles de reconnaissance vocale.

Un module d'interprétation plus flexible est nécessaire par rapport à ce que nous avons implémenté dans notre chaîne d'interaction. Nous nous orientons vers l'utilisation d'un outil d'analyse morpho-syntaxique, pour étendre et rendre plus flexible notre grammaire d'interprétation.

3.6.1 Discussion

Les ontologies présentent une solution très intéressante pour améliorer et affiner la connaissance de l'environnement par le robot. Pour l'instant, seulement des études préliminaires ont été menées avec la création d'une ontologie primitive, le point bloquant étant la création d'un module d'interprétation plus fin en sortie de la reconnaissance vocale. Celui-ci doit être suffisamment flexible pour approcher le langage naturel. Un outil d'analyse morpho-syntaxique combiné avec une grammaire d'interprétation pourrait permettre une analyse plus fine de ce qui est énoncé par le locuteur.

3.7 Conclusion

Dans ce chapitre nous avons présenté notre système d'interaction avec le robot, permettant de répondre à la problématique de recherche d'objets. Nous avons formalisé notre architecture générale d'interaction homme-robot composé de trois parties : la chaîne de perception, le gestionnaire d'interaction, et la chaîne de réponse. Nos contributions scientifiques se situent principalement autour de l'amélioration de l'interaction en tenant compte du contexte.

Ainsi, nous avons proposé une système générique de fusion bayésienne de plusieurs systèmes. Celui-ci est illustré à partir de deux micros et 2 moteurs de reconnaissance. Ce dispositif permet de prendre en compte la distance entre l'utilisateur et chaque micro pour sélectionner le système probablement le meilleur à chaque instant.

Nous proposons aussi un système de retour visuel (feedback) permettant de compléter la synthèse vocale pour une clarification de l'interaction, et enfin nous présentons des pistes pour améliorer la gestion de l'interaction en fonction du contexte grâce à la représentation des connaissances sous forme d'ontologies.

De nombreuses perspectives sont envisageables, en particulier au niveau de l'interprétation des hypothèses de reconnaissance. Nous pourrions utiliser des outils d'analyse morpho-syntaxique pour extraire des informations supplémentaires des hypothèses. Nous pourrions aussi utiliser les N-Bests hypothèses de reconnaissance et les confronter à notre système d'interprétation pour choisir la plus probable et ainsi renforcer chaque moteur de reconnaissance.

Des améliorations peuvent aussi être apportées à notre système de fusion en filtrant le signal d'entrée des micros pour une amélioration globale du SNR. Enfin de nouveaux détecteurs d'activité vocale pourraient réduire le WER en effectuant une segmentation plus précise. Par exemple, il pourrait tenir compte de zones de silence plus grandes pour éviter la sur-segmentation du signal.

Chapitre 4

Scénarii robotiques

4.1 Introduction

Ce chapitre se focalise sur l'intégration des modalités et des développements présentés dans les chapitres précédents, à savoir, le détecteur d'intentionnalité et l'interaction multimodale. Nous avons intégré ces composantes à un scénario d'interaction homme-robot cohérent avec l'objectif final du projet, c'est-à-dire qu'il permet une interaction avec l'usager pour une tâche d'aide à la recherche d'objets égarés dans un espace privatif appartenant à une personne âgée atteinte de troubles cognitifs légers.

Dans le cadre du projet RIDDLE, l'environnement n'est pas instrumenté, nous devons alors choisir une plateforme robotique embarquant des capteurs extéroceptifs permettant une vaste variété de capacités perceptuelles tout en étant peu intrusive dans l'environnement de l'usager. Ainsi, le robot se trouvant dans une pièce de vie, il doit être capable de se déplacer dans la pièce de façon autonome. De plus, comme il doit interagir avec l'utilisateur de la manière la plus naturelle possible, il faut qu'il soit équipé d'un micro et de haut-parleurs pour pouvoir répondre à l'utilisateur. D'autres dispositifs de feedback sont aussi à envisager comme la communication non-verbale avec le robot. Enfin, comme la finalité du projet RIDDLE est un robot d'assistance à domicile, le robot doit être facilement acceptable pour l'utilisateur. Cette acceptabilité peut être déterminée par la morphologie même du robot (aspect physique avenant, taille, etc.)(Johal et al., 2014), mais aussi par son comportement (réactivité, proactivité, gestes trop rapides/lents, etc.).

Dans la grande majorité des scénarii d'interaction homme-machine, l'utilisateur et le robot sont directement placés face à face et commencent à interagir, la « non-interaction » n'étant pas forcément gérée. Peu de robots d'assistance intègrent des comportements proactifs. Ce chapitre propose donc un scénario complet comprenant des comportements proactifs et prenant en compte une phase de pré-interaction appelée *monitoring* où le robot attend des informations verbales et/ou non-verbales pour démarrer une conversation en anticipant les besoins de l'utilisateur.

Ce chapitre se compose d'un état de l'art dans la section 4.2. Dans la section 4.3, nous faisons un descriptif des avantages et inconvénients des plateformes communément utilisées pour des scénarii d'interaction au LAAS-CNRS. Le scénario robotique envisagé est décrit dans la section 4.4. Ses

implémentations sur les plateformes du LAAS-CNRS sont décrites dans la section 4.5. Enfin leurs mises en œuvre lors de campagnes d'acquisition sont présentées dans la section 4.6. Finalement nous concluons quant au choix des plateformes et des scénarii dans la section 4.7.

4.2 Etat de l'art

Dans cette section, nous présentons un état de l'art portant sur les différentes applications de robotique d'assistance à la personne et les plateformes généralement utilisées pour cela. Nous exposons aussi les travaux existants liés aux robots proactifs se rapprochant de notre application.

4.2.1 Robotique d'assistance médicale

Tout d'abord, il est nécessaire de faire le point sur les plateformes utilisées dans la robotique d'assistance médicale pour les personnes âgées. Celle-ci étant en plein essor, de plus en plus de travaux de recherche ont été conduits ces dernières années. Cette augmentation se justifie par le contexte social d'une population vieillissante, et d'un personnel aidant de plus en plus rare. La grande majorité de ces travaux n'a pas pour vocation de remplacer les aidants, mais bien de les assister dans un ensemble de tâches automatisables.

Dans leurs travaux (Xiong et al., 2007), les auteurs présentent un robot permettant d'aider les personnes à se lever de leur siège, qu'elles soient âgées ou paraplégiques. Ce robot est conçu exclusivement pour cette tâche et ne ressemble donc pas à un robot humanoïde. Une fois la personne levée, il peut la porter tout en se déplaçant. Un algorithme de calcul de trajectoire, basé sur des réseaux de neurones, a été implémenté pour le déplacement. Ce robot est très proche d'une chaise roulante automatisée permettant en plus de maintenir l'utilisateur debout. Son originalité réside dans le fait que l'utilisateur peut alors exécuter des tâches qui lui seraient normalement inaccessibles avec ce genre de handicap. Ce robot est donc conçu dans une optique de manutention de la personne, et il permet de rendre l'usager un peu plus autonome dans sa vie de tous les jours.

Kumar et al. présentent aussi un robot de type « chaise roulante » (Kumar et al., 2013). Cette fois-ci le robot est équipé d'une tablette et d'un bras articulé permettant d'attraper des objets. La manipulation des objets est apprise par démonstration, et un écran permet à l'utilisateur de parler et donner des ordres au robot. L'une des nouveautés de ces travaux est le pilotage du robot par commandes vocales.

Dans la même veine, Wei et al. présentent une méthode de contrôle du robot « Elderly assistant and Walking assistant » (Wei et al., 2012) basée sur un capteur tactile de glissement. Leur méthode de contrôle se base sur quatre « ordres tactiles » : avancer, reculer, tourner à droite, tourner à gauche. Ce robot sert à la fois de déambulateur, mais aussi de chaise roulante suivant les capacités de la personne âgée.

Le plus souvent, les robots pré-cités sont conçus pour répondre à une tâche très spécifique dans le domaine médical tout en ayant une possibilité d'évolution quasiment nulle pour d'autres tâches. Cela est principalement dû à leur forme : un robot chaise roulante pourra difficilement balayer une pièce sans modification majeure de son architecture matérielle. De plus dans toutes ces expériences,

l'interaction se fait au moyen d'écrans tactiles ou bien de capteurs sensitifs. Dans le projet RIDDLE, les capacités perceptuelles requises pour l'interaction ne permettent pas l'usage de robots spécialisés pour l'assistance médicale des personnes. De plus, dans l'optique d'une interaction aussi naturelle que possible, un robot ressemblant trop à une machine risque fort d'être rejeté par une personne âgée. Enfin, la dernière difficulté concernant les robot pré-cités se situe dans leur façon d'interagir. En effet, ceux-ci sont non-proactifs : ils attendent un ordre de l'utilisateur. Dans un cadre d'interaction naturelle, le projet RIDDLE se focalise sur la pro-activité du robot, tout en limitant l'intrusion que l'utilisateur pourrait alors ressentir.

4.2.2 Robotique d'assistance anthropomorphique

Cet état de l'art continue sur la robotique d'assistance se rapprochant plus de notre domaine d'application puisqu'il concerne les robots d'assistance anthropomorphiques. En effet, le but ultime du projet RIDDLE étant une implémentation sur le robot Roméo, nous étudions ici l'usage de tels robots dans la littérature.

Dans leurs travaux (Yamazaki et al., 2012), les auteurs définissent la robotique d'assistance pour les personnes âgées sur trois niveaux : l'assistance industrielle (robots d'accueil, nettoyage, etc.), l'assistance dans le domaine de la santé (robot d'aide au déplacement, robot de dépistage, etc.), et enfin, la robotique d'assistance à domicile (entretien de la maison, robots compagnons). Les auteurs exposent également quelques scénarii robotiques principalement axés sur la manipulation d'objets et embarqués sur le robot HRP2 et le robot AR (Assistant Robot). Ces travaux appartiennent à la robotique d'assistance à domicile, le projet RIDDLE s'inscrivant lui-aussi dans cette même catégorie.

Cette fois, en robotique de service, les auteurs Han et al. présentent une architecture embarquée sur le robot Tiro (Han et al., 2009). Dans ces travaux, le robot est utilisé comme assistant d'enseignement pour enseigner la musique à des enfants. L'enseignant peut diriger le robot à l'aide d'un écran tactile en cas de défaillance de la reconnaissance vocale. Ces travaux sont prometteurs pour l'acceptabilité du robot, ce qui peut s'expliquer par le visage et la petite taille de la machine évoquant un jouet.

En terme d'acceptabilité, les auteurs présentent cinq situations d'interaction avec le robot Nao dans (Johal et al., 2014). Ces situations sont : l'enseignement, le jeu, la protection, le réconfort et enfin le coaching. Ces travaux montrent que l'acceptabilité d'un tel robot dépend non seulement de son aspect physique, mais aussi de la façon dont il est utilisé. Ainsi, le robot Nao est extrêmement bien accepté en tant que partenaire de jeu, alors qu'il aura plutôt un effet négatif s'il est utilisé pour réconforter l'utilisateur.

Avec les robots anthropomorphiques, nous remarquons que la variabilité des applications augmente grandement. Cela est principalement dû au nombre de capteurs embarqués, qui rendent ces robots très polyvalents, et donc particulièrement indiqués pour le projet RIDDLE. Dans le cadre de celui-ci, nous nous sommes donc orientés vers un robot anthropomorphique. Enfin, un robot d'assistance ressemblant même grossièrement à un être humain est plus susceptible d'être toléré par un utilisateur non-expert : ceci paraît important pour une application à domicile. Cependant, dans la majorité des applications les robots attendent une action explicite de l'utilisateur pour agir.

4.2.3 Pro-activité

Ainsi, dans cette section, nous nous centrons sur les travaux liés aux robots proactifs présents dans la littérature.

Dans (Schmid et al., 2007), les auteurs décrivent une architecture permettant d'anticiper les besoins d'un utilisateur. Les actions sont principalement orientées pour une assistance dans la cuisine : ranger les plats, nettoyer, servir de l'eau, etc. Le scénario commence directement par une phase d'interaction où l'utilisateur demande au robot d'aller chercher un pot contenant de l'eau. Le robot décide ensuite que faire de ce pot en fonction des ces observations : le poser sur un plateau, servir un verre d'eau, etc. Cette architecture proactive est formalisée par un réseau bayésien permettant de déduire une « probabilité d'intention », un peu à la manière de notre détecteur d'intentionnalité dans un contexte de coopération homme-robot.

Toujours en ce qui concerne la proactivité, les auteurs Pinheiro et al. présentent une architecture cognitive pour robots pro-actifs dans (Pinheiro et al., 2010). Cette fois-ci les travaux sont centrés sur la planification pro-active de tâches dans le cadre de la coopération homme-robot. Le robot doit adapter sa façon d'interagir avec l'utilisateur. La planification est modélisée par un réseau de neurones permettant cet apprentissage. Cette architecture est embarquée sur un robot anthropomorphique équipé de deux bras articulés pour la manipulation d'objets.

De la même façon, dans (Buss et al., 2011), les auteurs utilisent le concept de proactivité pour récupérer des informations manquantes pour une application de dialogue homme-robot en langage naturel. Ils fusionnent un détecteur d'émotions, des informations extraites de la reconnaissance vocale et un algorithme de reconnaissance de gestes. Dans ces travaux, le scénario consiste en un robot demandant son chemin à un interlocuteur humain. Le robot avance directement vers un utilisateur et commence à lui poser des questions. Ce scénario a été validé par une étude sur utilisateurs (user study) d'un échantillon de 6 personnes non-expertes.

Dans (Pandey et al., 2011), les auteurs présentent un scénario où l'utilisateur doit donner un objet au robot, ou le rendre atteignable. Lors du scénario, le robot demande à l'utilisateur de lui tendre un objet et peut avoir deux attitudes distinctes. Une attitude proactive : le robot tend le bras pour attraper l'objet, ou une attitude non proactive : le robot ne bouge pas. Ainsi, lors de cette étude sur utilisateur, les personnes recrutées trouvent l'attitude du robot déstabilisante lorsque le robot n'est pas proactif, validant l'idée lorsque ce dernier n'est pas proactif, nous confortant dans l'idée qu'un robot proactif permettrait une interaction plus naturelle.

Enfin les travaux (Gorur and Erkmen, 2014) utilisent la notion de proactivité dans le contexte d'anticipation des mouvements de l'utilisateur pour l'évitement d'obstacles. Ainsi, le robot planifie à nouveau ses déplacements en fonction de l'intention exprimée par l'utilisateur (par exemple, aller s'assoir devant l'ordinateur). Il n'y a cependant aucune réelle interaction entre le robot et l'utilisateur.

Ainsi, très peu de travaux existent sur la proactivité dans le sens « anticipation des besoins de l'utilisateur ». De plus, d'après nos connaissances, la majorité des travaux alliant interaction homme-robot et proactivité démarrent toujours directement lors de la phase d'interaction proximale. Nous souhaiterions donc innover ici en proposant un robot proactif dans un scénario complet où le robot commence par une situation de *monitoring* statique loin de l'utilisateur et hors-interaction, pour

continuer dans un état d'interaction proximale avec l'utilisateur.

4.3 Plateformes robotiques du laboratoire

Dans le cadre du projet RIDDLE, le choix d'une plateforme robotique est primordial dans la conception d'un scénario et la compatibilité des algorithmes avec les capteurs embarqués sur le robot. Dans cette section, nous présentons les différentes plateformes du LAAS-CNRS envisagées pour l'implémentation des scénarii RIDDLE, et en avons choisi quelques-unes. Les capacités, architectures logicielles, avantages et inconvénients de chaque plateforme sont résumés dans la table 4.1.

TABLE 4.1 – Comparatif des plateformes robotiques du LAAS-CNRS.

Robot	**Archi.**	**Capteurs**	**Avantages** **Inconvénients**
Rackham (roues) figure 4.1a	Genom	camera LadyBug, laser 2D, ceinture à ultrasons, caméra RGB	possibilité d'ajouter des capteurs facilement, facilité d'emploi, architecture logicielle et matérielle obsolète
PR2 (roues) figure 4.1b	ROS	laser 2D au sol et à balayage, deux bancs stereoscopiques, capteur Kinect	puissance de calcul embarquée, facilité d'emploi, plateforme très complète, communauté active, très demandé
Nao (humanoïde) figure 4.2a	NaoQi, passerelle ROS	micros, cameras RGB, capteurs tactiles	facilement transportable, facilité d'emploi, Choregraphe, déplacements difficiles
HRP2 (humanoïde) figure 4.2b	OpenHRP, passerelle ROS	caméras RGB, possibilité d'ajout de Kinect	humanoïde, puissance de calcul, pas de micros, 3 personnes minimum pour les expériences
Roméo (humanoïde) figure 4.2c	NaoQi, passerelle ROS	caméra RGB, capteur X-Tion, capteurs tactiles	choregraphe, design, facilité d'emploi, marche non disponible

4.3.1 Capteurs et architectures matérielles

Dans un premier temps, nous avons listé les capteurs indispensables pour faire fonctionner les différentes modalités implémentées dans le projet RIDDLE. Nous avons absolument besoin d'un

capteur RGB-D : un capteur Kinect ou X-Tion, et au moins un microphone pour l'intégration de l'interaction.

Ainsi, nous pouvons déjà exclure le robot Rackham des robots car il n'embarque pas de microphone. Le PR2 ne possède pas de micro directement intégré, mais l'installation d'un capteur Kinect permet d'avoir une entrée audio. L'éventuelle fixation d'un capteur Kinect sur le robot HRP2 pourrait aussi résoudre le problème. Enfin, les plateformes Nao et Roméo intègrent nativement plusieurs micros.

Enfin, les robots à roues sont naturellement plus faciles à manœuvrer que les robots humanoïdes puisque le risque de chute est inexistant. Un robot humanoïde nécessitera donc plusieurs personnes lors des expériences pour sécuriser le robot et aider au déplacement du portique le soutenant. De plus, comme le projet RIDDLE n'est pas axé sur les mouvements de déplacement du robot, nous ne nous appliquons pas à développer une solution de déplacement propre, mais nous utiliserons plutôt des solutions déjà implémentées qui ne sont en général pas disponibles sur les robots humanoïdes. De part leur taille, à l'exception du robot Nao, les robots humanoïdes sont donc globalement plus contraignants en terme de mise en œuvre.

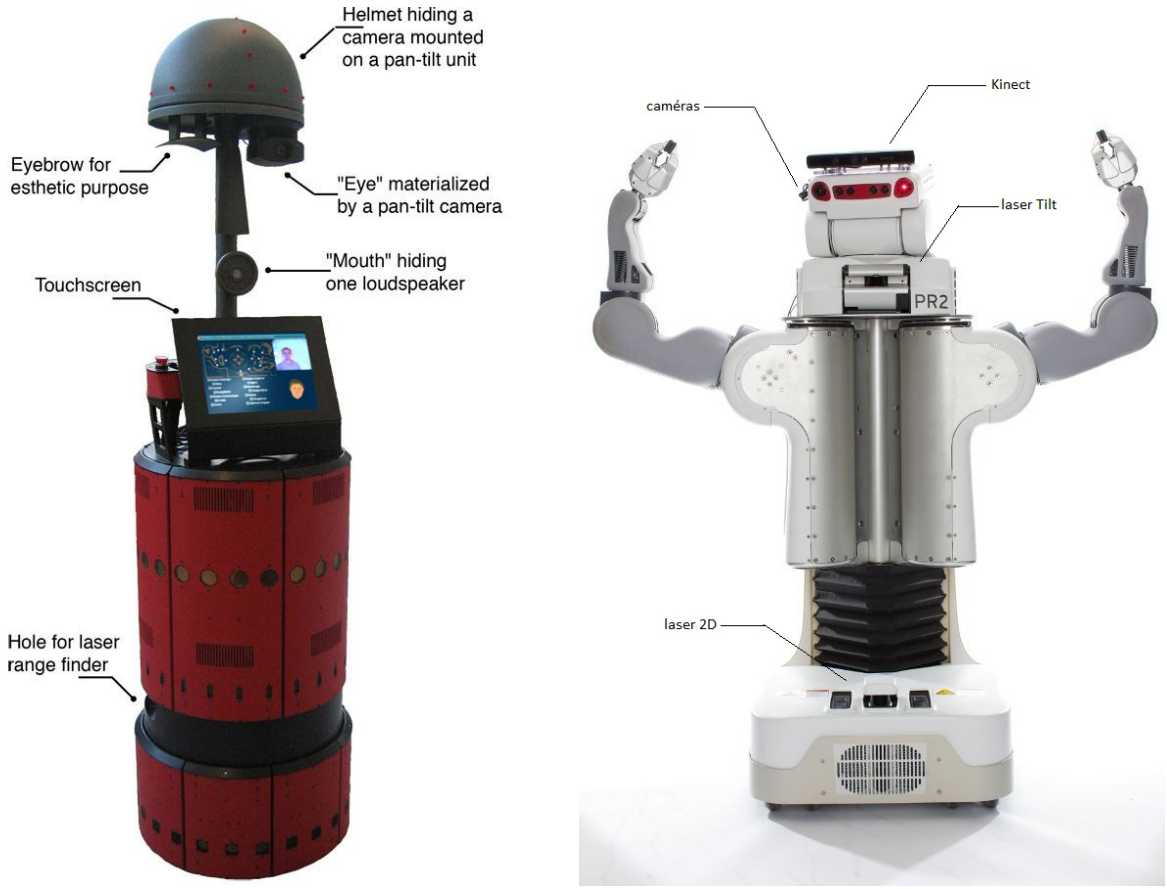
4.3.2 Architecture logicielle

En nous focalisant sur l'architecture logicielle, nous nous rendons compte que Genom (Fleury et al., 1997) et (Mallet et al., 2002), conçue par le LAAS-CNRS possède quelques modules qui ne sont plus à jour avec les versions plus récentes de Linux, rendant cette architecture obsolète pour les applications de perception.

NaoQi est l'architecture créée pour les robots de la société Aldebaran Robotics. Elle permet notamment l'utilisation du logiciel Choregraphe permettant une programmation graphique simplifiée pour le développement d'un grand nombre d'applications de bases, telles que les déplacements du robot, la reconnaissance vocale, etc. Le seul point faible de cette architecture vient du fait qu'elle n'est pas Open-Source et que son utilisation est limitée aux détenteurs de robots Aldebaran.

Enfin, la dernière architecture que nous présentons ici est ROS [1]. Celle-ci a été développée par Willow Garage, la société ayant implémenté OpenCV et conçu le robot PR2. Comme NaoQi, ROS permet une programmation par modules. Cependant, cette dernière est beaucoup plus axée sur la robotique expérimentale et intègre notamment un grand nombre d'outils de visualisation et de mesure, ainsi qu'un format de fichier permettant d'enregistrer et rejouer l'intégralité d'une expérience de robotique : les **rosbags**. La création de corpus est ainsi grandement facilitée, et les expériences peuvent être rejouées à l'identique ou bien en modifiant certains algorithmes. Un autre avantage non négligeable de cette architecture vient aussi de sa communauté très large et active, à la fois dans le monde et au sein du laboratoire, permettant un développement rapide. Cette architecture ne se limite pas à une utilisation dans le domaine robotique seul puisqu'elle permet facilement de créer un environnement de calcul déporté sur plusieurs ordinateurs. De nombreuses passerelles existent, permettant notamment de l'utiliser avec l'architecture NaoQi.

1. http ://www.ros.org/



(a) Rackham (b) PR2

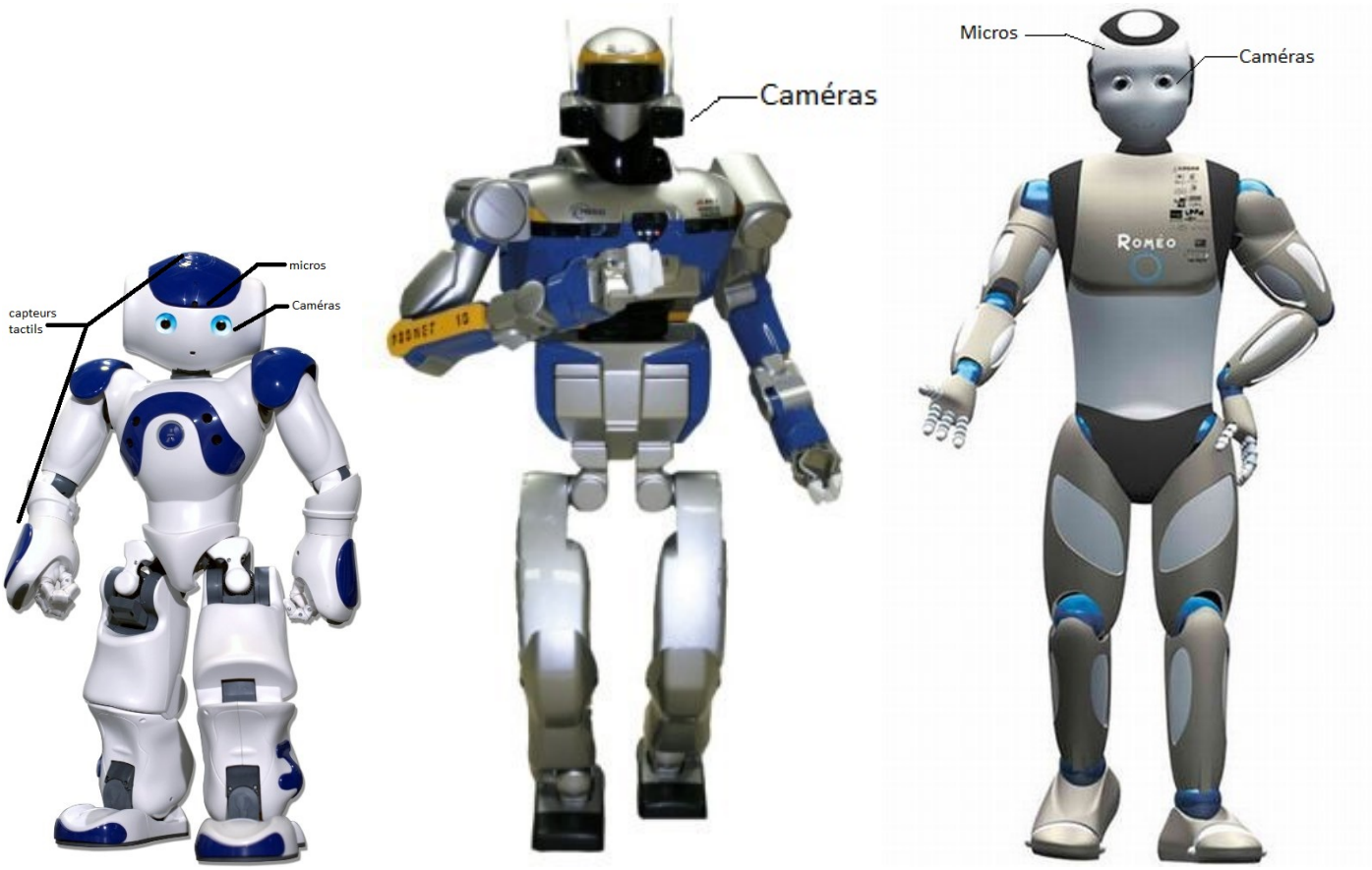
FIGURE 4.1 – Plateformes robotiques non-anthropomorphiques présentes au LAAS-CNRS.

4.3.3 Choix des plateformes

Considérant ces différentes plateformes, nous en avons sélectionné trois. Nao, la première, nous a permis de prendre en main l'architecture NaoQi qui est identique à celle du robot Roméo, tout en permettant un transport facile pour des acquisitions au gérontopôle de Toulouse. Ce robot était une bonne solution pour l'intégration en attendant la livraison du robot Roméo, et il a servi à la réalisation d'une première campagne d'acquisition.

Nous nous sommes ensuite orientés vers le robot PR2. De par l'architecture ROS et le capteur Kinect embarqué, il possède tous les pré-requis pour la conception du scénario RIDDLE global présenté dans la section 4.4. De plus, une passerelle existant entre les architectures ROS et NaoQi, la grande majorité des développements pourront être portés rapidement sur le robot Roméo.

Enfin, le robot Roméo est toujours la plateforme ciblée pour la fin du projet RIDDLE. La marche n'étant pas fonctionnelle sur le robot, l'intégralité du scénario ne peut pas être jouée. Pour l'instant l'accent est mis sur le portage des capacités d'interaction et le détecteur d'intentionnalité.



(a) Nao (b) HRP2 (c) Romeo

FIGURE 4.2 – Plateformes robotiques anthropomorphiques présentes au LAAS-CNRS.

4.3.4 Bilan sur les plateformes

Lors de l'étude des différentes solutions offertes par les robots du laboratoire, nous avons pu remarquer plusieurs tendances générales dans la communauté robotique. Bien que de nombreux efforts soient fournis dans l'intégration de caméras et capteurs de qualité, la gestion audio est très souvent mise à l'écart ou traitée de façon partielle. Par exemple, les micros de Nao et Romeo se situent à côté du ventilateur de refroidissement de la tête, ce qui les rend pratiquement inutilisables. Certains robots en sont même dépourvus.

Au niveau de l'architecture logicielle embarquée, une tendance semble se dégager avec l'utilisation de plus en plus importante de ROS. L'utilisation de NaoQi risque de rester très marginale puisqu'elle est limitée à une utilisation sur les robots d'Aldebaran. Enfin, Genom semble être de moins en moins maintenu et semble limité à une utilisation au sein du LAAS-CNRS.

Considérant ces différents aspects, nous nous sommes orientés tout d'abord vers le robot Nao. En effet, le but du projet étant une intégration sur Roméo, une prise en main de cette architecture

le plus tôt possible nous a semblé intéressante. Nous nous sommes ensuite décidés à effectuer des implémentations sur le PR2. Celui-ci permettant d'être opéré simplement et possédant une communauté très active. Nous avons gagné en temps de développement. De plus, une passerelle (bridge) existante entre ROS et NaoQi, nous pourrons envisager un portage simple des fonctionnalités sur le robot Roméo à partir du moment où il sera disponible.

4.4 Scénario RIDDLE

Dans le cadre de cette thèse, nous avons imaginé un scénario principal. Nous commençons par rappeler le cahier des charges du projet RIDDLE.

4.4.1 Cahier des charges

Le projet repose sur plusieurs hypothèses :
- Tous les scénarii sont mono-utilisateur. Une seule personne se trouve dans l'environnement du robot, dans les champs de vision des caméras et à portée de microphones.
- Un certain nombre d'objets *a priori* connus du robot par apprentissage visuel se trouvent dans l'environnement du robot.
- Le projet RIDDLE n'est pas axé sur les mouvements du robot. Il n'y a donc aucun développement particulier sur la manipulation d'objets et les mouvements de marche.
- À terme l'environnement visé est le domicile de l'utilisateur composé d'une pièce de vie principale où le robot doit se trouver.
- Le robot se trouvant chez l'usager, il ne doit pas être intrusif mais doit éventuellement pouvoir anticiper les besoins de l'utilisateur. Nous nous orientons donc vers un robot pro-actif non-intrusif.
- Pour une interaction plus naturelle, le robot doit pouvoir s'adapter à l'utilisateur, et avoir un temps de réaction relativement faible.

4.4.2 Scénario général

À partir de ce cahier des charges nous avons construit un scénario principal pour le projet en accord avec le gérontopôle de Toulouse, partenaire du projet RIDDLE.

Ce scénario se compose de 4 étapes clefs :

Home-Tour : dans un premier temps, le robot ne connait par l'appartement dans lequel il est placé. Le « Home-Tour » peut être assimilé à une étape d'initialisation où le robot va apprendre la carte de l'environnement non-connue *a priori* , puis va localiser les objets connus dans sa base de données une première fois pour les placer dans cette carte. Cette étape peut éventuellement être répétée pour tenir compte des changements survenant dans l'environnement. En effet, un appartement étant un environnement humain, il présente une grande variabilité : déplacement de chaises, modification de la décoration, déplacement d'objets, nouveaux obstacles, etc.

Monitoring : une fois l'initialisation effectuée, le robot se déplace dans sa zone de garage définie en accord avec l'utilisateur. Cette zone permet de limiter le sentiment d'intrusion vécu par

l'utilisateur et d'assigner au robot une zone de charge fixe au cours du temps. Depuis cette zone, le robot peut ainsi observer les activités de l'utilisateur à distance, cette étape pouvant être assimilée à de la vidéo-surveillance. Pour répondre au besoin de pro-activité, c'est aussi à ce moment là que nous utilisons le détecteur d'intentionnalité présenté dans le premier chapitre. Celui-ci permet de détecter l'intention de communication de l'utilisateur, il rend le robot pro-actif à un éventuel besoin, puisque c'est lui qui pourra initier une conversation.

Transitions : Cette phase repose sur le mouvement du robot. C'est le seul moment (hors initialisation) où le robot se déplace dans l'environnement. D'une part, cette étape consiste à passer de la phase de *monitoring*, à la phase d'interaction proximale. Le robot doit non seulement se déplacer vers l'utilisateur tout en évitant les obstacles, mais aussi se positionner par rapport à l'usager pour une interaction de qualité.

Interaction proximale : Dans cette phase, le robot se situe proche de l'utilisateur (distance d'interaction inférieure à 2m) et démarre une interaction essentiellement vocale avec l'utilisateur. C'est durant cette étape que l'utilisateur peut poser des questions au robot dans une situation d'interaction homme-machine. Dans le contexte RIDDLE, cette étape est centrée sur la recherche d'objets. Une fois l'interaction interrompue par l'utilisateur, le robot retourne dans sa position de garage pour jouer le scénario à nouveau.

L'originalité de ce scénario réside principalement dans la phase de *monitoring*. En effet, la plupart des travaux de recherche sur l'interaction homme-machine ont tendance à démarrer directement l'interaction par notre dernière phase du scénario. Nous présentons ainsi un scénario un peu plus global, permettant de gérer tout ce qui se situe avant (et après) l'interaction proximale. Ce scénario est résumé dans la figure 4.3. De cela découle une deuxième contribution pour la création d'un robot pro-actif non-intrusif. En effet, la majorité du temps, celui-ci se trouve dans un coin de la pièce (non-intrusif) mais utilise des informations non-verbales pour démarrer une phase d'interaction (pro-activité, voir la section 1.4).

Pour pouvoir réaliser ce scénario, nous avons besoin d'un certain nombre de modalités :

Home-Tour : Lors de cette phase, nous avons besoin d'un algorithme de type SLAM permettant à la fois de cartographier l'environnement, de segmenter les surfaces de rangement potentielles et éventuellement permettre la localisation du robot dans l'espace. Cet aspect est principalement géré par la société Magellium et n'est donc pas abordé ici, excepté pour l'intégration robotique. La détection d'objets est aussi nécessaire pour pouvoir mettre à jour la position des objets dans la carte de l'environnement précédemment construite.

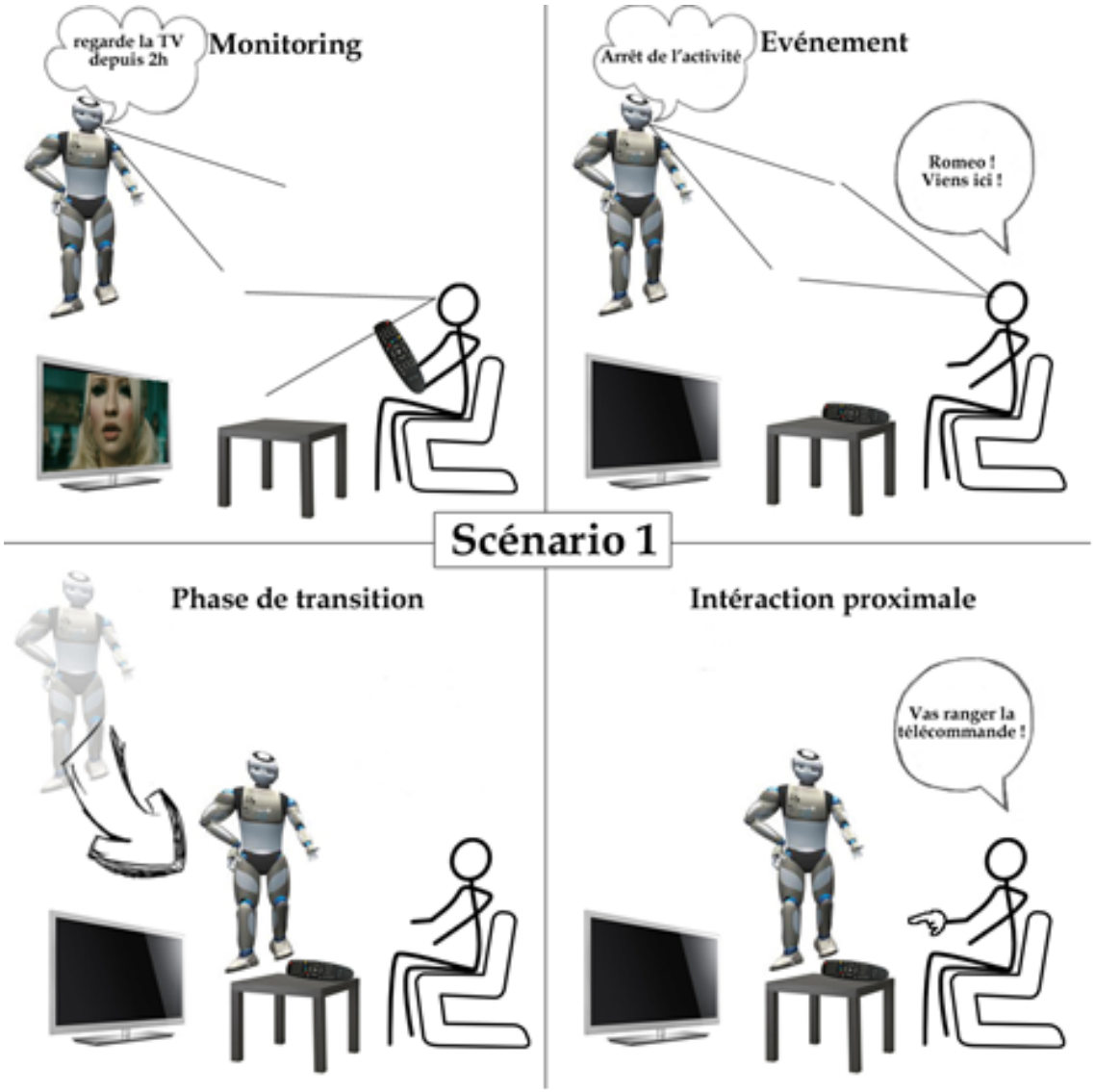


FIGURE 4.3 – Illustration des trois grandes étapes du scénario RIDDLE (*monitoring* à l'interaction proximale).

Monitoring : Durant cette phase, le robot doit détecter la personne pour vérifier qu'elle se trouve dans la pièce et diriger ces capteurs vers elle (caméras, Kinect, etc.). Ensuite le module d'intentionnalité est initialisé et attend que l'utilisateur déclenche une détection. Ce module d'intentionnalité se base sur la détection d'orientation de visage, d'orientation des épaules, et la détection d'activité vocale. Un module de reconnaissance de gestes peut aussi être utile pour que l'utilisateur puisse joindre le geste à la parole pour démarrer une conversation.

Transition : Pour cette phase, aucune modalité n'est vraiment nécessaire. la plupart des algorithmes de déplacements simples étant déjà présents nativement sur le robot. Nous avons ainsi utilisé des modules pré-existants sur le robot.

Interaction proximale : Enfin, pour la dernière phase, les APIs de reconnaissance vocale sont nécessaires ainsi qu'un module permettant de gérer l'interaction. Un dispositif de feedback visuel est aussi utile pour clarifier les différentes attitudes du robot.

Toutes les modalités sont résumées dans la table 4.2.

TABLE 4.2 – Modalités nécessaires pour l'implémentation du scénario RIDDLE.

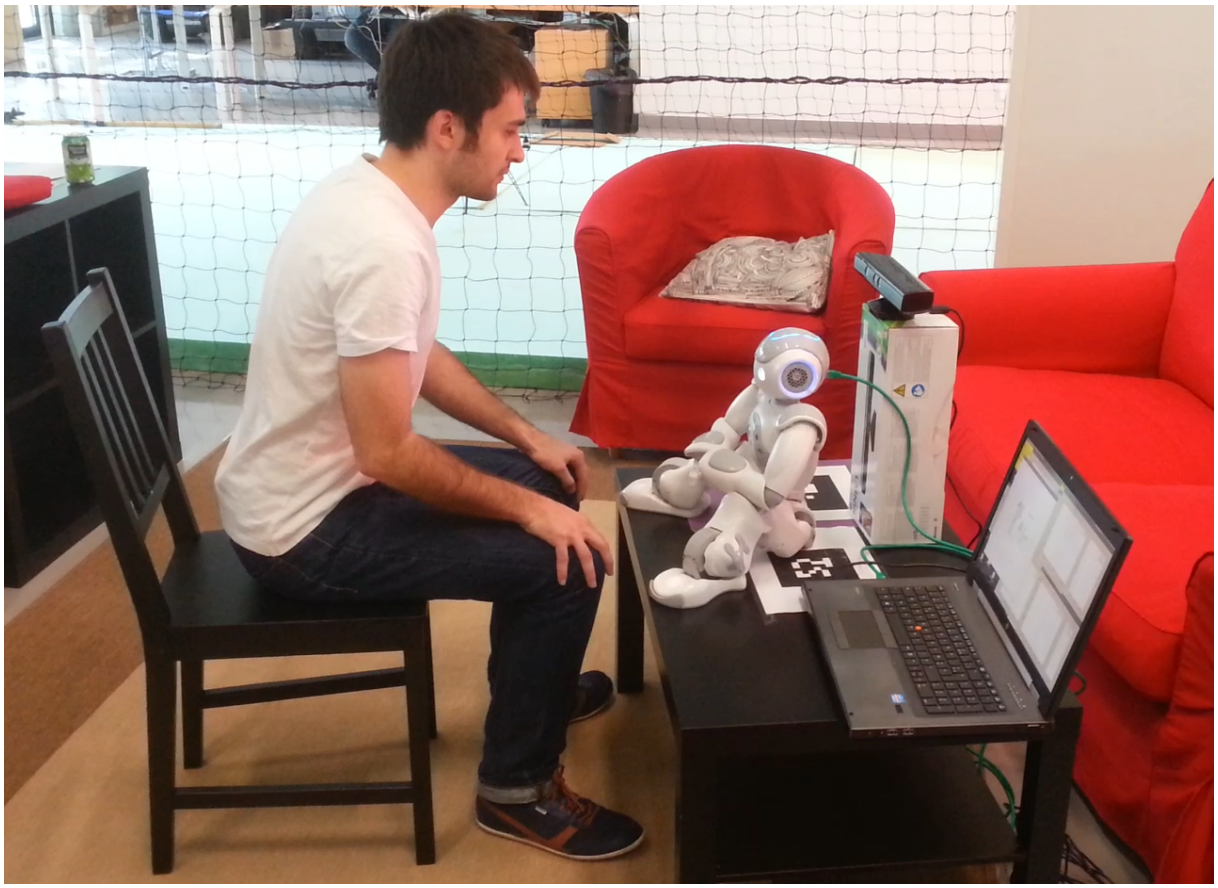
Modalité	**Phase**	**capteurs extéroceptifs**
Cartographie de l'environnement	Home-Tour	Kinect
Détection d'objets	Home-Tour	Kinect
Détection de visages	Monitoring, Transition	Capteur RGB (Kinect)
Détection d'orientation de visages	Monitoring	Kinect
Détection des épaules	Monitoring	Kinect
Activité vocale	Monitoring, Interaction	Micros
Reconnaissance de gestes	Monitoring	Kinect
Filtrage visuel basé PSO	Monitoring	Kinect
Intentionnalité	Monitoring	Kinect, Micros
Interaction	Interaction	Micros
Feedback visuel	Interaction	Aucun
Supervision	Toutes	Aucun

4.5 Mise en œuvre du scénario sur les plateformes

Cette section décrit les implémentations réalisées sur les robots NAO et PR2. Ces implémentations ont servi à faire des démonstrations dans le cadre du projet RIDDLE, et ont été alimentées par deux campagnes d'acquisitions pour valider nos développements et extraire des informations permettant d'améliorer et étoffer notre scénario. Ces campagnes ont été faites en partenariat avec le gérontopôle de Toulouse pour une validation du point de vue médical.

4.5.1 Première implémentation : robot Nao

Ce scénario est axé sur l'implémentation d'une première version du détecteur d'intentionnalité et une reconnaissance vocale basée sur PocketSphinx. Une situation d'interaction est présentée sur la figure 4.4.



(a) Interaction

(b) Intentionnalité détectée

(c) Pas d'intentionnalité

FIGURE 4.4 – Situation d'interaction dans le scénario I. Le bouton d'allumage du torse sert de feedback visuel pour connaître l'état de la détection d'intentionnalité.

Dans ce scénario, Nao est installé sur une table face à l'utilisateur. L'utilisateur commence par vaquer à ses occupations en étant assis devant le robot, par exemple, il joue sur son téléphone.

TABLE 4.3 – Modalités du scénario RIDDLE implémentées sur le robot Nao.

Modalité	**Phases**	**Capteurs extéroceptifs**	**États**
Cartographie de l'environnement	Home-Tour	Kinect	N,A
Détection d'objets	Home-Tour	Kinect	Nul
Détection de visages	Monitoring, Transition	Capteur RGB (Kinect)	N,A
Détection d'orientation de visages	Monitoring	Kinect	**Intégré**
Détection des épaules	Monitoring	Kinect	**Intégré**
Activité vocale	Monitoring, Interaction	Micros	**Intégré**
Reconnaissance de gestes	Monitoring	Kinect	N,A
Filtrage visuel basé PSO	Monitoring	Kinect	N,A
Intentionnalité	Monitoring	Kinect, Micros	**Intégré (version préliminaire)**
Interaction	Interaction	Micros	**Intégré (version préliminaire)**
Feedback visuel	Interaction	Bouton du Torse	Intégré
Supervision	Toutes	Aucun	Nul

Lorsqu'il requiert l'assistance du robot, il se tourne vers lui (figure 4.4a) et lui formule sa requête. Le détecteur d'intentionnalité rentre alors en jeu en mesurant l'orientation de l'utilisateur à l'aide du capteur Kinect fixé préalablement sur le robot, ainsi que son activité vocale à l'aide du détecteur de PocketSphinx. La détection azimutale de locuteur fournie dans l'architecture logicielle NaoQi a aussi été utilisée comme information sur la position de l'utilisateur par rapport au robot Nao. Une fois l'intentionnalité détectée (figure 4.4b), le robot commence à poser des questions en fonction des informations fournies par l'utilisateur. Lorsque l'objet à trouver est identifié, le robot cherche sa position dans une base de données enregistrée *a priori* . Si l'objet est présent dans la base, le robot indique sa position relative à l'utilisateur. Les modalités implémentées sur ce robot vis-à-vis du scénario global sont présentées dans le tableau 4.3.

Le robot Nao a été choisi comme première solution pour une démonstration. Cela se justifie par son environnement de développement identique à celui de Romeo, ciblé comme plateforme

finale du projet. De plus, les gériatres l'ont trouvé pertinent pour une bonne acceptabilité par les personnes âgées. Ainsi, pour toutes ces raisons, ce robot nous a paru être une bonne solution pour une première série de démonstrations et une première campagne au gérontopôle. D'autant plus qu'il possède l'atout non-négligeable d'être facile à transporter.

Cependant, ce robot n'a pu être une solution de développement robotique sur le long terme, principalement dû à sa taille. En effet, il est impossible pour Nao de se déplacer dans la pièce, tout en ayant un champ de vue à hauteur du visage de l'utilisateur, et donc, de réaliser la partie « Home-Tour » du scénario. Cela rend aussi la plupart des algorithmes de détection de visage et d'estimation d'orientation inopérants, puisque la plupart sont conçus pour fonctionner de façon frontale.

4.5.2 Deuxième implémentation : robot PR2

Pour cette deuxième implémentation, nous avons choisi le robot PR2 conçu par Willow Garage. Comme il est de taille humaine, il n'y aura pas de problème pour le fonctionnement des algorithmes de vision par ordinateur. De plus, c'est un robot sur roues, il ne présente donc pas de problèmes de stabilité et l'architecture ROS embarque nativement un certain nombre d'algorithmes permettant de planifier son déplacement. Enfin, depuis la création d'une passerelle ROS sous NaoQi, tous les développements effectués sur le robot PR2 seront intégrables sur le robot Roméo.

Dans cette implémentation du scénario RIDDLE, nous nous sommes focalisés sur la phase de *monitoring* jusqu'à la phase d'interaction proximale décrite dans la section 4.4. L'ensemble du déroulement du scénario est supervisé par l'API Smach, permettant la construction d'une machine à états pour la planification des différentes actions du robot. Ce scénario intègre les étapes du *monitoring* à l'interaction proximale.

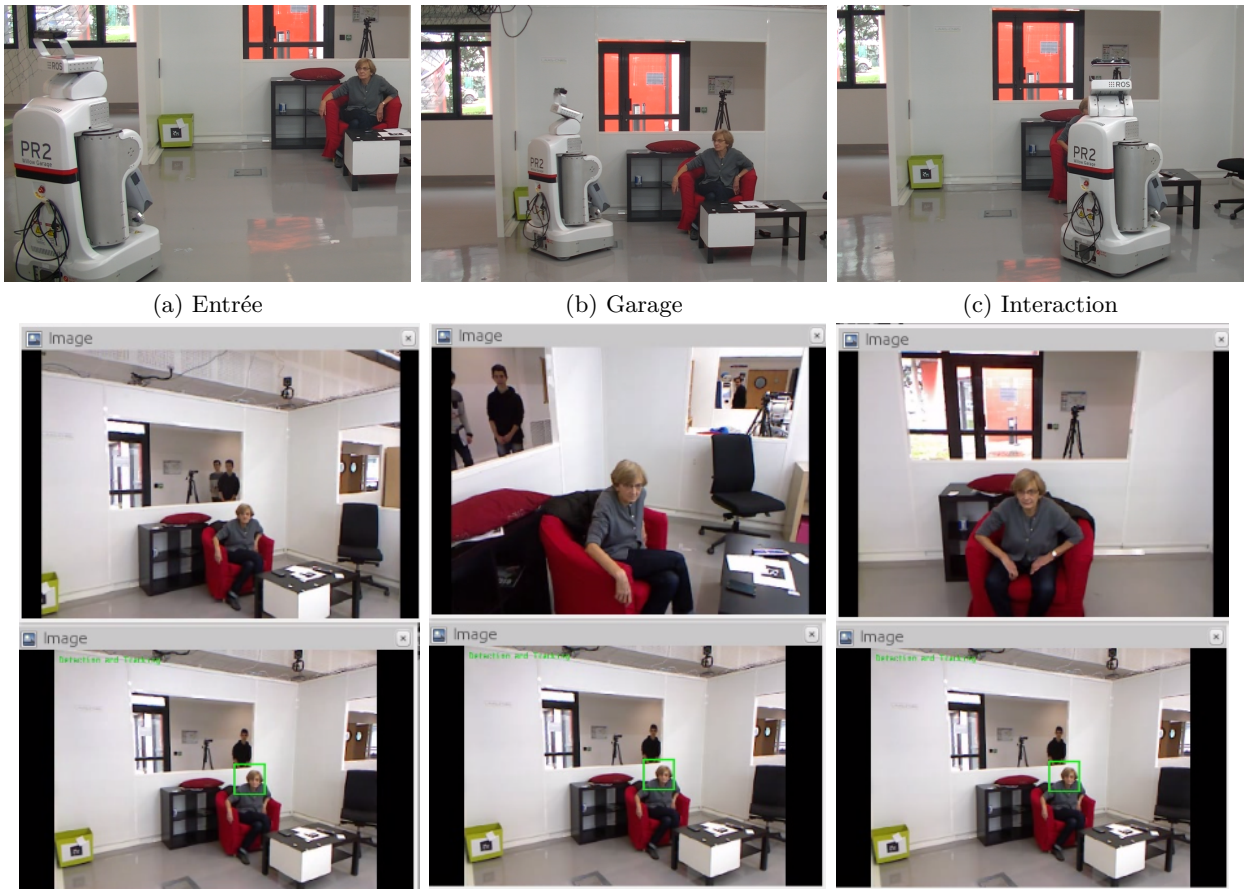
Avant la première étape de *monitoring*, le robot commence par entrer dans la pièce où se déroule l'expérience et cherche l'utilisateur (voir les figures 4.5a et 4.5d). Cette détection permet de vérifier que celui-ci est bien présent dans la salle. Une fois celle-ci effectuée et le robot garé, il entre dans sa phase de *monitoring* et enclenche le détecteur d'intentionnalité (voir les figures 4.5b et 4.5e). Enfin, lorsque l'intentionnalité est détectée, celui-ci commence la phase d'interaction proximale (voir les figures 4.5c et 4.5f). Une fois la session d'interaction terminée, le robot retourne se garer et le scénario peut recommencer.

Nous proposons sur le PR2 un scénario presque complet d'interaction en environnement humain pour une utilisation autonome par un utilisateur non-expert. Ce scénario a été validé lors d'une campagne d'expérimentations effectuée dans l'appartement du bâtiment ADREAM du LAAS-CNRS. La campagne est présentée dans la section 4.6.2, les photos présentées sur la figure 4.5 permettent de visualiser les étapes 1, 3 et 5, et sont extraites du corpus acquis durant cette campagne.

Les développements et intégrations nécessaires pour les différentes phases du scénarii sont résumés ci-dessous :

Monitoring : pour cette phase, un détecteur d'utilisateur a été utilisé pour repérer l'utilisateur dans la pièce. La détection d'intentionnalité a aussi été améliorée avec l'ajout du filtre basé PSO présenté dans le deuxième chapitre.

Transition : le déplacement du robot a été implémenté dans le scénario à l'aide de l'architecture



(a) Entrée (b) Garage (c) Interaction

(d) Vue du robot de l'entrée (e) Vue du robot du garage (f) Vue du robot en interaction

FIGURE 4.5 – Positions et vues du robot durant le scénario joué lors de la campagne présentée dans la section 4.6.2. Le cadre vert correspond au détecteur d'utilisateur.

ROS. Nous avons ainsi prévu plusieurs phases de transitions où le robot PR2 se déplace d'un endroit à un autre de la salle tout en évitant les obstacles présents sur son chemin. À la fin de chaque déplacement, le robot se réaligne avec l'utilisateur pour le conserver dans son champ de vision.

Interaction : pour la phase d'interaction, nous avons implémenté une application Android se comportant comme un noeud ROS, permettant ainsi de récupérer les buffers du micro du smartphone. Celui-ci étant proche de l'utilisateur, cela permet d'avoir une signal sonore avec un SNR exploitable. En effet, dans l'implémentation sur Nao, l'utilisateur se trouvait proche du robot. Ici, le robot étant potentiellement plus loin de l'utilisateur, le smartphone est utilisé comme une alternative pour acquérir le signal audio. La gestion de l'interaction a aussi été intégrée dans l'environnement Smach, et le moteur de reconnaissance alternatif Google Speech API a été implémenté comme alternative à PocketSphinx.

L'avancement des développements par rapport au scénario global, présenté dans la section 4.4, est récapitulé dans la table 4.4.

TABLE 4.4 – Modalités du scénario RIDDLE implémentées sur le robot PR2.

Modalité	**Phases**	**Capteurs extéroceptifs**	**États**
Cartographie de l'environnement	Home-Tour	Kinect	**En développement**
Détection d'objets	Home-Tour	Kinect	Nul
Détection de visages	Monitoring, Transition	Capteur RGB (Kinect)	**Intégré**
Détection d'orientation de visages	Monitoring	Kinect	**Intégré**
Détection des épaules	Monitoring	Kinect	**Intégré**
Activité vocale	Monitoring, Interaction	Micros	**Intégré**
Reconnaissance de gestes	Monitoring	Kinect	**En développement**
Filtrage visuel basé PSO	Monitoring	Kinect	**Intégré**
Intentionnalité	Monitoring	Kinect, Micros	**Intégrée**
Interaction	Interaction	Micros	**Intégrée, ajout du smartphone Android**
Feedback visuel	Interaction	Raspberry (chapitre 3)	Nul
Supervision	Toutes	Aucun	**Intégré**

Intégration

L'utilisation de l'outil Smach a permis de formaliser notre implémentation présentée sur la figure 4.6 sous la forme d'une machine à états présentée sur la figure 4.7.

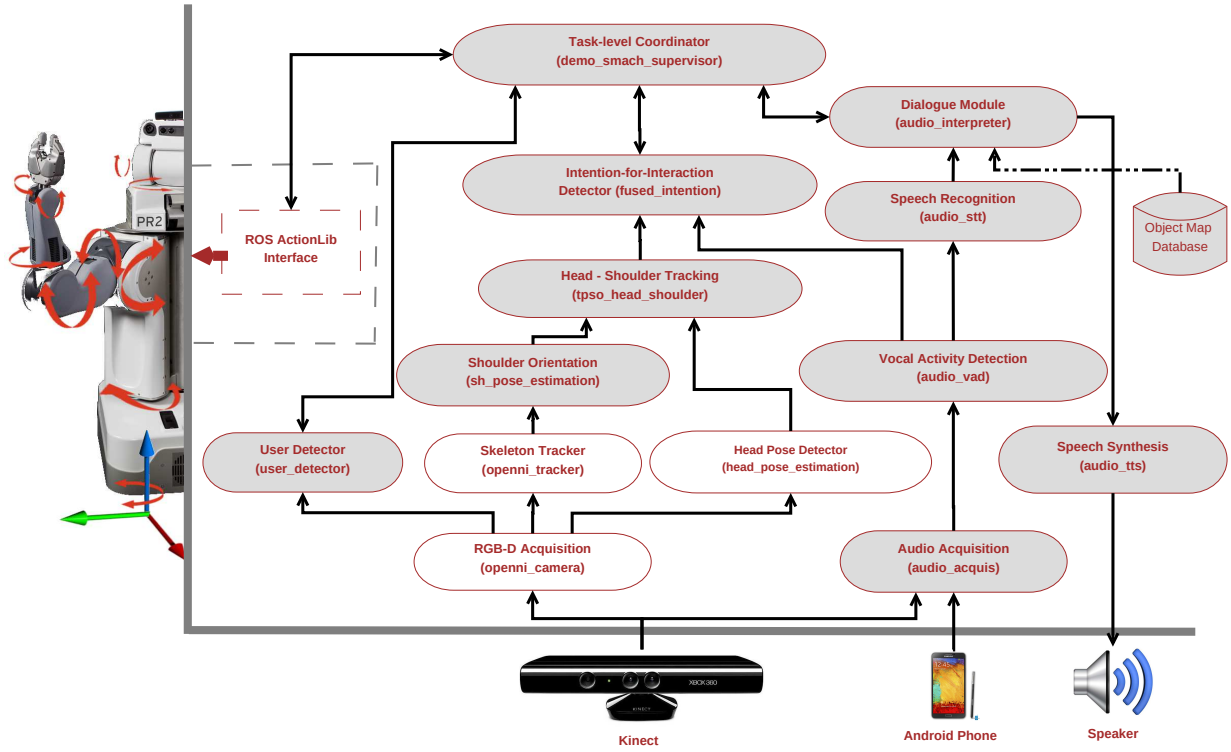


FIGURE 4.6 – Illustration du système embarqué sur le PR2 sous l'architecture ROS. Chaque rectangle arrondi représente un nœud ROS. Les flèches correspondent aux messages échangés entre les nœuds. Les zones grisées représentent les nœuds que nous avons développés, tandis que les zones blanches sont uniquement des adaptations de l'existant.

Ainsi la figure 4.6 est une représentation des modules principaux du scénario embarqué sur le PR2. Au milieu, nous pouvons voir la partie gérant l'intentionnalité avec le module « intention-for-interaction ». À droite, tout ce qui est lié à l'interaction, et tout en haut, Smach, représenté par le module de supervision. Le scénario complet est présenté sur la figure 4.7. Nous observons alors les modules « FINDPERSON », « INTENT4INTERACT » et « DIALOGUE » qui représentent respectivement la détection d'utilisateur, le module d'intentionnalité, et la gestion de l'interaction.

FIGURE 4.7 – Visualisation du système construit à l'aide de Smach. Chaque zone grisée est une « sous-machine à états ». Les sorties sont représentées par les ovales rouges.

4.5.3 Implémentations futures

A l'heure où nous écrivons ces lignes, le projet RIDDLE entame sa phase finale, et le scénario est en cours de complétion pour atteindre le maximum d'objectifs fixés en début du projet. Nous présentons ici les développements en cours pour le scénario final du robot PR2, ainsi que le portage d'une partie du scénario sur le robot Roméo.

robot PR2

Le principal incrément du scénario embarqué sur PR2 est l'ajout d'une phase de « home tour » pour initialiser les connaissances du robot. Lors de cette étape, celui-ci fait le tour de la pièce en construisant la carte de l'environnement à l'aide d'un algorithme de SLAM RGB-D fourni par la société Magellium. Cet algorithme permet aussi de segmenter les surfaces planes visibles dans l'environnement du robot. Après que cette carte a été créée, le robot va entamer un nouveau tour dans la pièce pour détecter les objets dont il connait le modèle *a priori* et les placer dans la carte. Des précisions pourront être apportées verbalement par l'utilisateur grâce au module d'interaction ou bien en cas d'ambiguïtés sur la détection.

Des améliorations vont aussi être apportées quant à la détection d'intentionnalité à l'aide d'un module de reconnaissance de gestes, permettant ainsi de tenir compte d'éventuels mouvements de l'utilisateur. Ce module de reconnaissance a déjà été développé mais n'a pas été intégré et testé sur le robot, et a donné lieu à une publication dans la conférence ACM : Audio Mostly (Pellegrini et al., 2014).

Une dernière démonstration prévue début décembre 2015 permettra de clore le projet RIDDLE avec le scénario complet embarqué sur le PR2.

robot Roméo

Le robot Roméo ayant été livré au LAAS-CNRS, nous envisageons aussi d'utiliser la passerelle ROS vers NaoQi pour porter la phase de détection d'intentionnalité jusqu'à la phase d'interaction sur le robot. Le robot ne pouvant se déplacer seul, nous ne pourrons malheureusement pas jouer le scénario complet implémenté sur le PR2.

4.6 Campagnes d'acquisition

Durant cette thèse, les implémentations présentées dans la section précédente ont donné lieu à plusieurs démonstrations, une campagne d'acquisition, et une campagne d'expérimentations. Nous détaillons les deux campagnes au sein de cette section.

4.6.1 Campagne d'acquisition au Gérontopôle

La première campagne d'acquisition a eu lieu au gérontopôle du CHU de Casselardit à Toulouse. Celle-ci a eu pour but de valider la liste des objets à cibler dans nos scénarii de recherche, tout en observant la façon de communiquer des utilisateurs lors des demandes d'aide au robot. Un

autre objectif de cette campagne a été d'enregistrer les conversations avec les personnes âgées pour éventuellement apprendre le vocabulaire utilisé dans ce genre de situations et adapter les modèles acoustiques aux voix des personnes âgées.

Disposition de la salle

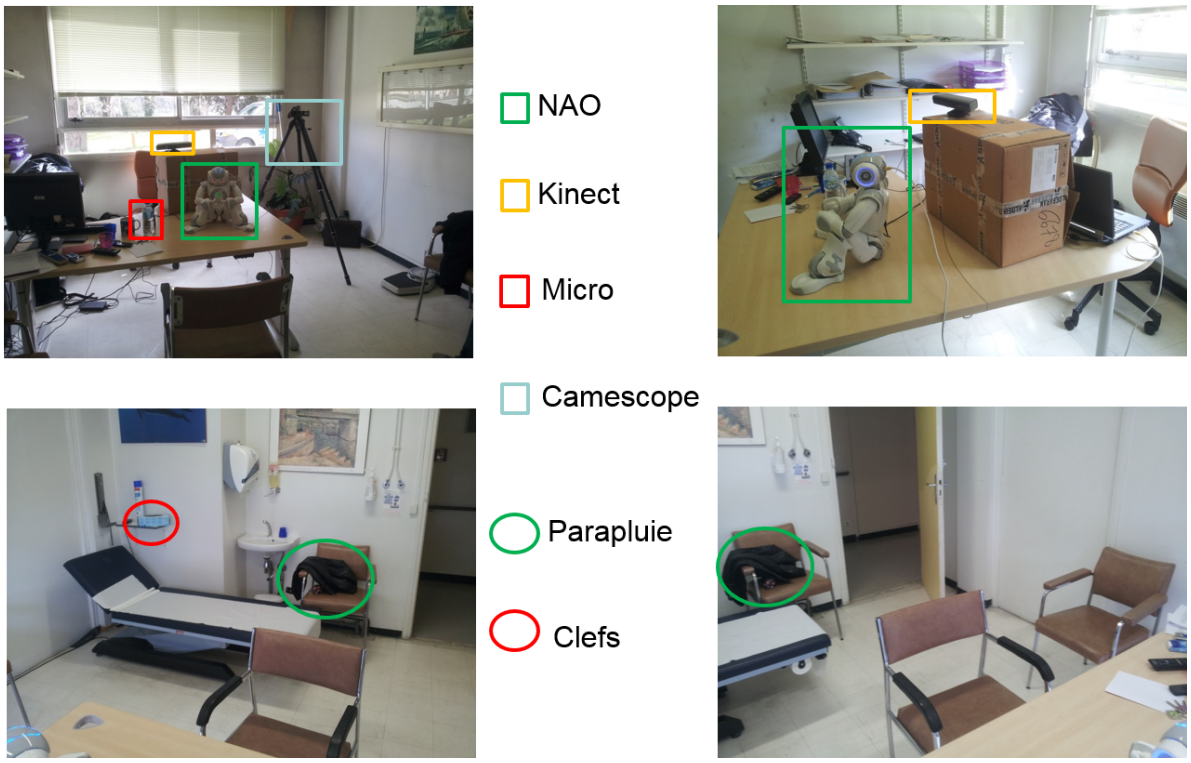


FIGURE 4.8 – Disposition de la salle lors de la première campagne d'acquisition au gérontopôle de Toulouse.

L'organisation de la pièce est résumée sur la figure 4.8. Concernant la partie vidéo, un caméscope classique a été placé dans l'angle de la pièce (rectangle bleu) de façon à observer tout le déroulement des expériences. Un capteur Kinect (rectangle jaune) a aussi été fixé près du robot Nao, assis sur la table et orienté vers l'utilisateur (rectangle vert), de façon à enregistrer les images couleur et profondeur lors de l'interaction face au robot.

Pour la partie audio, le caméscope a permis de faire une prise de son dans la pièce. Un micro fourni par Aldebaran Robotics ayant les mêmes caractéristiques que les micros de Romeo a été placé proche des personnes interrogées pour simuler la qualité d'enregistrement présents sur le robot. Un micro externe (rectangle rouge) relié à un ordinateur a été utilisé pour permettre une prise de son de meilleure qualité. Enfin, nous nous sommes servis du micro présent sur le robot Nao, car des microphones du même type sont embarqués sur le robot Roméo, ce qui nous a permis d'avoir un

TABLE 4.5 – Corpus acquis durant la campagne au gérontopôle de Toulouse.

Capteurs	Durée	Vol. données	Nb. personnes (F/H)
Caméscope	3h22m35s	31,19 Go	13(9/4)
Kinect	4h19m11s	41,51 Go	13(9/4)
Mic. classique	4h19m11s	2,25 Go	13(9/4)
Mic. Nao	2h35m44s	157 Mo	9(6/3)
Mic. Romeo	1h55m28s	1,19 Go	7(5/2)

aperçu de la qualité audio exploitable directement sur le robot.

Un parapluie a été caché de manière à être facile à trouver tandis qu'un trousseau de clefs a été mieux camouflé.

L'objectif de cette installation a été d'avoir le maximum de sources d'images et de pistes sonores suivant les solutions à adopter lors de futures implémentations, tandis que le gérontopôle a fait une synthèse de l'ensemble des objets égarés le plus souvent (Boudet et al., 2013; Boudet et al., 2014)

Protocole

La première session d'expérimentations a eu pour but de recueillir un certain nombre d'informations sur les habitudes des personnes âgées. Pour cela, nous avons reçu 13 personnes de plus de 60 ans dont 9 femmes et 4 hommes. L'ensemble des données audio comprend le micro embarqué sur le robot Nao, le micro du caméscope, le micro externe et enfin le micro équivalent à ceux présents sur le robot Romeo. Les données vidéos comprennent la flux RGB-D du capteur Kinect et la vidéo enregistrée par le caméscope. Les corpus enregistrés sont résumés dans la table 4.5. Nous avons créé un protocole en quatre phases répétées pour chaque personne interrogée.

Première étape : l'utilisateur commence par parler de son utilisation des objets présélectionnés. Cette étape a permis de déterminer le lexique minimal à prendre en compte pour la construction des grammaires. Elle a duré entre 10 et 30 minutes suivant l'éloquence de la personne interrogée.

Deuxième étape : l'utilisateur doit trouver un objet caché dans la pièce sans aide du robot. Cette phase a permis d'obtenir des informations sur la façon dont les personnes âgées se déplacent pour chercher des objets, exploitables en particulier par le corps médical, mais aussi en cas de création d'algorithmes d'apprentissage visuels et/ou sonores pour détecter une activité de recherche.

Troisième étape : l'usager doit trouver un objet caché dans la pièce. Cependant, l'utilisateur doit cette fois demander de l'aide au robot. Le robot se lève ensuite pour indiquer l'endroit où se trouve l'objet à la fois oralement et par un geste du bras. Durant cette campagne, le robot est piloté en mode « magicien d'Oz ». Cette phase a permis d'enregistrer la façon dont les personnes âgées se sont adressées au robot pour lui demander de l'aide et nous a donné des pistes pour créer une vraie mesure d'intentionnalité.

Quatrième étape : Les utilisateurs doivent donner leur avis sur l'expérience, en répondant à un questionnaire ouvert. Cette étape a permis au gérontopôle d'extraire des statistiques sur l'acceptabilité du robot Nao.

Discussion

Cette première campagne a permis de soulever un certain nombre de points. Tout d'abord, lors de la demande d'aide au robot, certains utilisateurs ont eu tendance à l'appeler par son prénom ou bien par « robot », tandis que d'autres lui ont posé directement une question. La politesse de la question a aussi varié de « Est-ce que tu pourrais m'aider à trouver le parapluie s'il te plaît ? », à « Je ne trouve pas mon parapluie. ». Nous pouvons ainsi trouver plusieurs types d'énoncés : la première phrase étant un exemple d'énoncé direct à la forme interrogative. La deuxième, n'est grammaticalement pas une question, il s'agit d'une assertion au mode affirmatif, la question pouvant être sous-jacente ou interprétée comme telle : « je ne trouve pas mon parapluie, peux-tu le trouver ? ». Cette variabilité dans la façon de demander des informations au robot et d'initier une phase d'interaction a permis de valider la faisabilité d'un détecteur basé sur l'orientation du corps et du visage, et la mise en place d'une composante vocale dans la détection.

Parallèlement à nos investigations sur l'intentionnalité, les gériatres ont pour leur part étudié les questionnaires et les données vidéos que nous leur avons transmises. Cette étude a donné lieu aux publications (Boudet et al., 2013) et (Boudet et al., 2014) portant principalement sur les objets égarés par les personnes âgées.

4.6.2 Campagne d'expérimentations : ADREAM

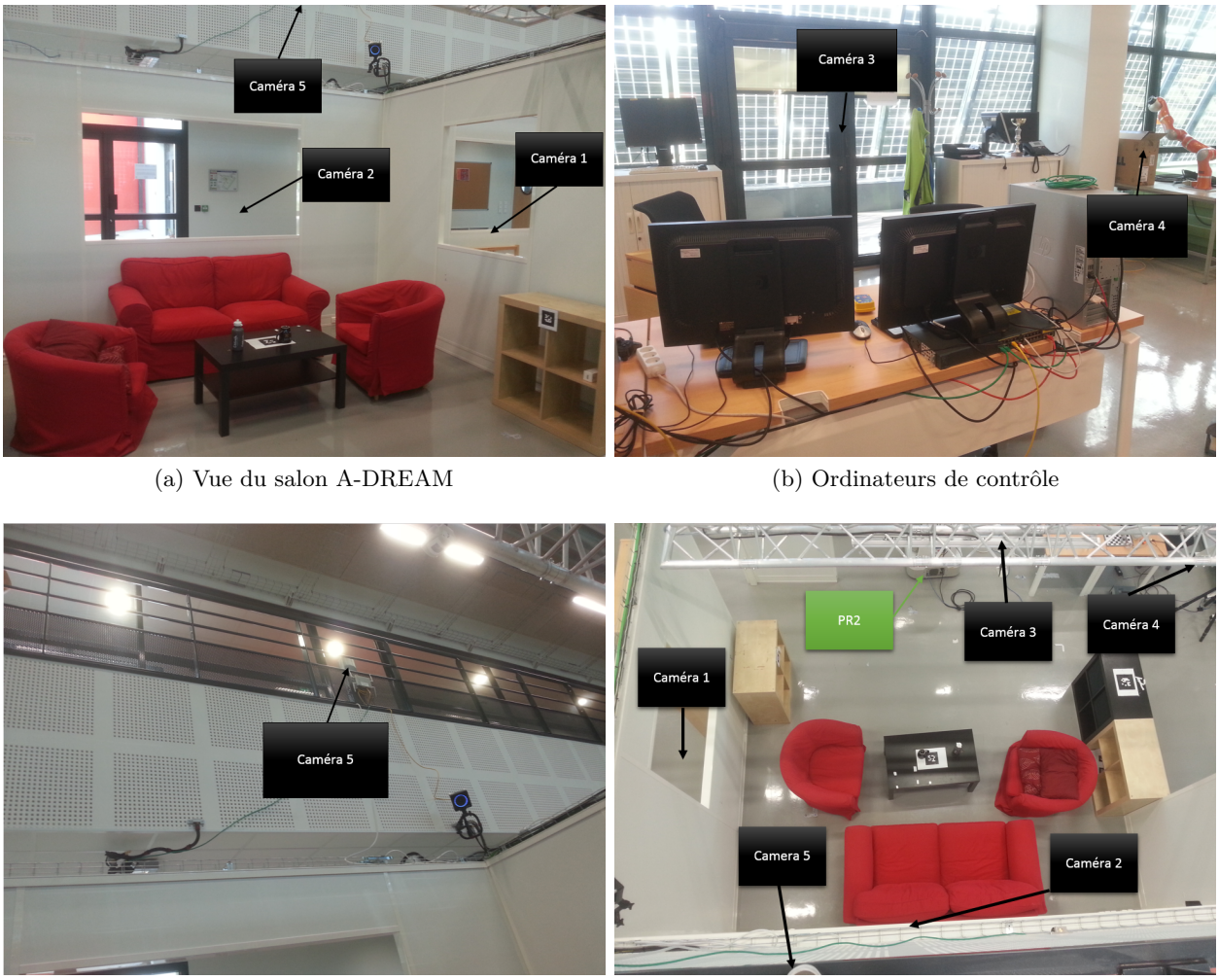
Le but principal de cette deuxième campagne a été de valider notre détecteur d'intentionnalité par des mesures sur des utilisateurs non-experts en robotique. Celle-ci a encore une fois été organisée en partenariat avec le gérontopôle de Toulouse. Contrairement à la campagne présente qui a consisté en une étude des réactions des utilisateurs face au robot piloté en mode magicien d'Oz, durant cette campagne, nous avons joué le scénario implémenté sur le PR2 et présenté dans la section 4.5.2. Aucune partie du scénario n'était effectuée en mode magicien d'Oz. Les positions des objets *a priori* présents dans la pièce d'expérimentation étant connues.

Les membres du gérontopôle ont quant à eux évalué l'acceptabilité du robot PR2 à l'aide de questionnaires à remplir à la fin de l'expérience.

Disposition de la salle

Lors de cette campagne d'acquisition, seule une pièce de l'appartement A-DREAM a été utilisée. Cinq caméras ont été disposées dans des endroits stratégiques de manière à pouvoir observer la scène de multiples points de vues. Celles-ci sont représentées sur les photos de la figure 4.9. Les caméras 1 à 4 ont été disposées sur des trépieds autour de l'appartement. La caméra 5 a été posée sur un trépied sur la mezzanine du premier étage pour avoir une vue globale du déroulement des expériences (voir la figure 4.9c).

Dans la pièce, deux assises ont été disposées près d'une table basse. L'utilisateur peut, au choix, s'asseoir sur un des sièges, ou bien rester debout. Le robot est situé hors des limites de la pièce pour simuler une entrée dans celle-ci au début de l'expérience. La position de garage (*monitoring*) se trouve à côté de l'un des sièges. Les objets sont préalablement cachés dans les différents meubles



(a) Vue du salon A-DREAM

(b) Ordinateurs de contrôle

(c) Mezzanine

(d) Vue de dessus

FIGURE 4.9 – Disposition des caméras dans l'appartement A-DREAM lors de la deuxième campagne d'acquisition.

de la pièce. Une liste de ces objets se trouve sur la table basse pour rappeler la liste des objets cibles aux utilisateurs.

Protocole

Cette deuxième session d'expérimentations a eu pour but de valider notre scénario tout en fournissant des statistiques à l'équipe du gérontopôle de Toulouse. Dans cette optique, la campagne s'est déroulée sur deux jours. Nous avons testé notre scénario sur 17 personnes de plus de 60 ans recrutées spécifiquement, soit 9 hommes et 8 femmes pour un total de plus de 2h d'enregistrement sous forme de rosbags (fichiers d'enregistrement de l'architecture ROS). Ces fichiers permettent de

TABLE 4.6 – Base de données acquise durant la campagne ADREAM.

	hommes (9)	**femmes (8)**	**total (17)**
durée	1h 13min 48sec	56min 48sec	2h 10min 37sec
taille rosbag 1	2,10Go	1,44Go	3,54Go
taille rosbag 2	763Mo	530Mo	1,26Go
taille rosbag 3	4,18Go	2,90Go	7,83Go

simuler les entrées de capteurs tels qu'ils ont réagi durant la campagne d'acquisition. Le rosbag 1 contient les données relatives au détecteur de personne et au détecteur d'intentionnalité. Il contient aussi toutes les matrices de transformations permettant de passer du repère monde aux différentes articulations du robot et à l'orientation de l'utilisateur. Le rosbag 2 a enregistré la partie audio avec les topics transmettant les buffers du smartphone Android et les buffers des micros du capteur Kinect. Enfin le rosbag 3 contient la configuration du capteur Kinect ainsi que la séquence vidéo associée. Les volumes et la durée des différents rosbags sont récapitulés dans la table 4.6. Le protocole d'expérimentation a été composé en trois étapes.

Première étape : avant l'expérience, les utilisateurs reçoivent des consignes sur le déroulement de celle-ci et les objectifs de cette mise en situation. Celles-ci sont dispensées par une personne du gérontopôle hors du périmètre de la pièce.

Deuxième étape : le robot se trouve à l'extérieur de la pièce. L'utilisateur commence par venir s'installer sur une des chaises présentes. Il peut aussi rester debout, du moment qu'il se trouve dans la pièce. Une liste des objets présents dans la pièce est posée sur la table, l'utilisateur pouvant demander au robot d'en trouver une sous-partie. Le scénario présenté en section 4.5.2 est alors joué.

Troisième étape : l'utilisateur doit ensuite remplir un questionnaire sur ses impressions sur le scénario qu'il vient d'effectuer. L'ensemble des questionnaire est actuellement en cours d'analyse par le gérontopôle.

Résultats et discussions

Cette deuxième session d'expérimentations a permis de soulever un certain nombre de points intéressants lors des phases d'interaction homme/robot, en plus des résultats liés à l'évaluation du scénario.

Ainsi, nous avons qualifié le scénario de « réussi », lorsque l'utilisateur est passé par toutes les états de la machine d'état créée à l'aide de l'outil Smach menant à un état final « succeeded » . Il est considéré comme « échoué » si la machine d'état se retrouve dans un des deux autres états : « aborted » ou « preempred ». Seule une réalisation du scénario n'a pas été menée jusqu'au bout. Nous obtenons ainsi un taux de succès de 94% sur un total de 17 personnes.

Concernant l'intentionnalité, dans **68,75%** des cas, elle a été détectée au premier essai de l'utilisateur et dans **18,75%** des cas lors du **deuxième essai**. Les utilisateurs restants n'ont réussi à déclencher le détecteur qu'à la troisième tentative. Seule une personne n'a pas réussi à déclencher le détecteur dû à une distance trop éloignée du robot, et donc à une défaillance de la modalité de

détection d'orientation de visage.

Ces conclusions reflètent ainsi les résultats obtenus lors de l'évaluation du détecteur d'intentionnalité en environnement maîtrisé, tout en validant la création d'un scénario robuste.

Une étude sur utilisateurs (user study) a été entamée par le gérontopôle de Toulouse et les premiers résultats ont permis d'étudier les réactions des utilisateurs vis-à-vis du système présenté. Ils ont ainsi appliqué une approche dite « bottom-up » en observant le comportement des usagers lorsqu'ils ont demandé de l'aide au robot pour trouver un objet perdu dans l'environnement. Sur les 17 personnes observées, 10 avaient déjà eu une expérience avec un robot. L'observation des expressions et du langage corporel des utilisateurs a donné un total de 11 expressions plutôt « souriantes », 7 « doutant » et 3 « en attente ». Sur les 17 volontaires, 11 se sont penchés vers le robot, qu'ils soient experts ou non. 5 utilisateurs sur les 17 ont eu tendance à parler différemment au robot dès le début de l'expérience, en ralentissant le débit de parole. Lorsque le robot a mal compris la phrase prononcée, les utilisateurs ont eu tendance à ralentir encore leur débit de parole, ou bien à hacher les mots, comme ils le feraient avec un humain ou un animal. Le point positif est que personne n'a manifesté de sentiment d'inquiétude par rapport à la présence du robot, ce qui a encore tendance à valider l'acceptabilité de notre scénario.

En plus de ces résultats, nous avons pu observer quelques défaillances, notamment au niveau de la gestion de l'interaction. Ainsi, certains utilisateurs ont commencé à parler avant que la reconnaissance vocale ne soit active, ou bien lorsque le robot allait leur parler. Ceci peut être expliqué par l'absence de feedback visuel sur le PR2 permettant d'indiquer à l'utilisateur quand parler. Ces résultats ont permis de lancer le développement du dispositif présenté dans le chapitre précédent. Ce feedback était présent sur le robot Nao, grâce à un ensemble de LEDs contrôlables intégrées dans le robot. Nous pourrons aussi utiliser les LEDs intégrées dans le torse du robot Roméo.

Un autre problème a été soulevé lorsque les utilisateurs ont haché les phrases. La détection d'activité vocale a alors sur-segmenté le signal audio, créant ainsi plusieurs phrases pour une seule effectivement prononcée. Une meilleur gestion de l'historique de la conversation permettrait de résoudre ce problème.

Enfin, bien qu'une liste d'objets ait été fournie aux usagers, certains ont employé d'autres termes. Par exemple, « boîte de Doliprane » au lieu de « boîte de médicaments ». Cela a révélé un problème de couverture lexicale de notre module d'interprétation qui est donc amené à évoluer et à intégrer plus de synonymes ou de formulations de phrases.

L'ensemble des travaux et expériences réalisées avec le robot PR2 a donné lieu à la soumission d'un article dans le revue CVIU qui est actuellement en révision.

4.7 Conclusion

Dans ce chapitre, nous avons présenté les implémentations réalisées sur les robots Nao et PR2, ainsi que celles à venir d'ici la fin du projet RIDDLE, sur les robots PR2 et Roméo. L'originalité de notre scénario réside dans le fait que celui-ci prend en compte toutes les phases de fonctionnement d'un robot à l'aide des étapes de *monitoring* et d'interaction proximale, permettant de mettre au point un robot à la fois pro-actif et non-intrusif.

Ces scénarii ont permis d'initier une campagne d'acquisition au gérontopôle de Toulouse et une campagne d'expérimentation dans l'appartement ADREAM du LAAS-CNRS, simulant un environnement humain. Cette deuxième campagne a permis de valider notre scénario et a été bien accueillie par les 17 volontaires experts et non-experts, sachant qu'aucune différence notable n'a été observée entre les deux groupes. Elle a aussi permis de vérifier le fonctionnement du détecteur d'intentionnalité et de soulever quelques nouvelles difficultés au niveau de l'interaction.

Ainsi de nouvelles pistes sont envisagées pour les futures démonstrations, avec l'installation du feedback visuel sur le robot PR2, l'élargissement de la couverture lexicale du module d'interprétation et enfin, la prise en compte plus fine de l'historique de l'interaction. Une étape « Home-Tour » sera aussi ajoutée pour initialiser les connaissances du robot et compléter le scénario implémenté jusque là.

Conclusion

Dans ce manuscrit, nous avons présenté nos travaux de recherche relatifs à la problématique de la perception de l'homme par un robot, dans l'optique d'une interaction. Ce travail de recherche et d'intégration, à la fois considérable et transverse, a été mené dans le but d'intégrer un ensemble de modalités perceptuelles et interactives afin de mettre au point un scénario robotique cohérent dans notre domaine d'étude. La robotique d'assistance étant une discipline éminemment polyvalente, nous nous sommes tournés vers des domaines aussi variés que passionnants tels que la vision par ordinateur (intentionnalité), l'optimisation, la reconnaissance automatique de la parole ou encore l'interaction homme-machine, l'informatique embarquée et l'électronique. Ce sujet d'étude touche également d'autres domaines tels que la psychologie cognitive, la linguistique ou encore la sémiologie. Ne faisant pas partie de notre champs d'expertise à proprement parler, ces domaines ont pu, à l'occasion, constituer des facteurs limitants dans le déroulement de nos travaux. Le manque de temps et de spécialisation dans ces dits domaines constitue très certainement un point d'amélioration possible au scénario de perception de l'homme dans une optique d'interaction homme-robot.

Nous avons fait le choix de découper le mémoire en quatre chapitres, chacun traitant une sous-partie des modalités implémentées dans les scénarii robotiques du dernier chapitre. Ainsi, le premier a été centré sur la détection d'intentionnalité, le deuxième sur la problématique de filtrage en suivi visuel, et le troisième sur l'amélioration de l'interaction dans un contexte de robotique d'assistance. Chaque chapitre a porté sur des modalités étroitement liées visant à créer un scénario robotique complet. Ces catégories ne peuvent donc pas être rendues étanches au regard de la mise en place du scénario final. C'est pourquoi nous avons jugé pertinent d'effectuer des aller-retours et rappels entre ces diverses catégories tout au long des quatre chapitres. En effet, la problématique d'interaction peut difficilement être disséquée et dissociée du scénario dans laquelle elle est mise en œuvre. De même, l'intentionnalité est une dimension intrinsèque de l'interaction, de même que la problématique de non-intrusivité nécessairement générée par un scénario d'assistance à domicile. De plus, la porosité du domaine robotique au sens large nous impose ce traitement scientifique global.

Cette problématique de recherche étant extrêmement vaste, nous considérons que notre scénario pourrait être grandement amélioré par de futures collaborations avec, par exemple, des linguistes, des chercheurs en psychologie cognitive ou en épistémologie, mais aussi des sociologues. En effet, la robotique étant destinée à un usage humain et qui plus est à une certaine génération, leur expertise pourrait apporter beaucoup à la versimilitude des scénarii.

Toutefois, bien que nous n'ayons pas eu les moyens ni le temps d'aller plus loin dans nos investigations, le bilan de nos travaux s'avère positif de par les résultats encourageants à beaucoup de nos expériences, notamment celles menées sur la détection d'intentionnalité, le filtrage et l'amélioration de l'interaction par la prise en compte d'éléments du contexte.

Ainsi, nous avons présenté un détecteur d'intentionnalité fondé sur l'exploitation de trois percepts visuels et construit à l'aide d'un modèle HMM. Cette information non-verbale révélatrice d'une intention d'interaction permet d'enclencher la phase d'interaction proximale du scénario robotique de manière proactive et surtout non-intrusive.

Dans le deuxième chapitre, nous avons exposé nos travaux scientifiques quant à la technique de filtrage par essaim de particules. Celle-ci est inspirée de l'algorithme d'optimisation PSO et permet

un filtrage des percepts visuels utilisés dans l'intentionnalité. De meilleurs résultats sont obtenus avec la mise en œuvre de cette technique qu'avec l'utilisation des algorithmes de filtrage particulaire.

Le troisième chapitre traite de l'amélioration par la prise en compte du contexte sonore et visuel du robot dans une architecture d'interaction. Celle-ci est notamment employée dans la phase d'interaction proximale du scénario RIDDLE. Les améliorations portent sur la création d'un dispositif de *feedback* visuel, ainsi que sur un système de fusion bayesienne pour l'amélioration de la chaîne de perception de la parole de l'utilisateur. D'autres travaux en cours sont exposés sur la représentation du contexte à l'aide d'ontologies.

Dans le dernier chapitre, nous présentons le scénario RIDDLE et ses implémentations sur deux plateformes robotiques : Nao et PR2. Ce scénario conçu en partenariat avec le gérontopôle de Toulouse à permis de programmer un robot autonome pour la tâche de robotique d'assistance à domicile. Ces développements ont été validés par diverses campagnes d'acquisitions et d'expérimentations sur des populations non-expertes.

Ces travaux ont tous donné lieu à des à des publications dans des conférences internationales : IEEE-ICIP (Mollaret et al., 2014) pour le filtrage, IEEE-ICME (Mollaret et al., 2015) pour l'intentionnalité, et une soumission dans la conférence IEEE-ICASSP pour l'amélioration de l'interaction. Un article soumis dans la revue CVIU est en cours de révision et porte sur les résultats de nos travaux sur l'ensemble du scénario du projet RIDDLE. Rappelons également que ces scénarii ont été validés par l'accueil de volontaires extérieurs au LAAS-CNRS pendant deux jours. Cette population d'utilisateurs pourtant non-experte en robotique a ainsi pu tester notre scénario qu'elle a accueilli favorablement.

Ce mémoire rend donc compte de nos contributions en matière de perception et de prise en compte du contexte dans le domaine extrêmement large qu'est la robotique d'assistance à domicile. Une part de l'originalité de ces travaux réside dans la compilation d'un grand nombre de modalités perceptuelles implémentées par nos soins mais aussi dans l'intégration de l'expertise des différents partenaires du projet qui sont Magellium, Aldebaran Robotics, l'équipe MINC du LAAS-CNRS et le gérontopôle de Toulouse. Ce travail collaboratif à permis de mettre en place un scénario robotique très complet et polyvalent, ce qui est nécessaire si nous voulons veut appréhender la complexité de l'environnement humain. L'ensemble des corpus employés lors de nos différentes expérimentations sont rendus disponibles sur demande, excepté les enregistrements des personnes âgées pour des raisons de droit à l'image [2].

Ces travaux ne s'achèvent pas avec la conclusion de ce mémoire : ce sujet est amené à être approfondi et de nombreuses pistes peuvent être explorées.

Ainsi, notre détecteur d'intentionnalité, bien que robuste, souffre parfois de défaillances dues à chacun des percepts visuels employés. Une amélioration de ceux-ci pourrait donc offrir de nouvelles perspectives pour les conditions d'utilisation du détecteur. Par exemple, un détecteur d'orientation du visage plus précis devrait améliorer l'efficacité de la perception de l'intentionnalité. De même, une intégration de percepts supplémentaires tels que de la reconnaissance de geste ou bien un détecteur d'activités de l'utilisateur pourrait étendre les possibilités interactives offertes par cette architecture. Pour l'instant, celle-ci est mono-utilisateur et ne permet pas de percevoir la présence

2. http ://homepages.laas.fr/cmollare/

de plusieurs personnes. Pour ce faire, il faudrait sans doutes créer un système de priorité : si deux personnes veulent interagir en même temps, vers laquelle le robot doit-il se diriger en premier ? L'ajout d'une telle fonctionnalité implique donc elle-même de nombreuses investigations, parmi lesquelles le problème de la ré-identification des utilisateurs.

Ce contexte multi-utilisateur implique également la modification de notre filtre par essaim de particules, car celui-ci devrait être étendu au suivi multi-cibles. Même dans un problème de suivi mono-cible, cette technique laisse de nombreuses perspectives d'évolution. Par exemple, nous pourrions envisager une auto-adaptation de la taille du nuage de particules à la dynamique courante de la cible. Cette technique est notamment employée dans le domaine du filtrage particulaire (Li et al., 2013). Nous pourrions aussi envisager un découpage hiérarchique du vecteur d'état permettant de modéliser un processus par parties. De plus, bien que nous ayons utilisé ce filtre dans un contexte de suivi visuel, rien n'empêcherait *a priori* de l'appliquer dans des domaines tels que le filtrage du signal audio.

L'architecture du module d'interaction pourrait ainsi être améliorée par l'utilisation d'un tel filtre. Celle-ci pouvant être très complexe, elle permet la plus grande amélioration. Il faudrait dans un premier temps rendre le dialogue plus souple et affiner l'interprétation de la parole de l'utilisateur et de son environnement. Nous pourrions envisager une technique d'apprentissage par renforcement pour la création du modèle de dialogue. L'interaction pourrait également être enrichie par l'intégration un module de reconnaissance et/ou de synthèse de gestes dans le robot. Une extension de l'interaction au contexte multi-utilisateurs pourrait aussi être envisagée. Cela impliquerait une modification de l'architecture en profondeur, notamment au niveau de la perception du signal. Il serait intéressant d'employer des algorithmes de localisation et de séparation de sources sonores pour dissocier la parole prononcée par les utilisateurs. Le dialogue devrait aussi pouvoir gérer la conversation de plus de deux personnes en modifiant la gestion de tours de parole (Kondo et al., 2012). En effet, dans notre contexte, les tours de parole sont alternés seulement entre le robot et l'utilisateur. Cette interaction est cependant rendue difficile dans un contexte robotique du fait des microphones embarqués sur les robots. En effet, peu d'efforts sont pour l'instant réalisés pour améliorer le rapport signal sur bruit de ceux-ci, résultant en l'intégration de capteurs premier prix. Cependant, une alternative à l'intégration de tels microphones se présente avec la démocratisation des *Smartwach*. En effet, celles-ci pourraient servir de microphones embarqués directement sur les utilisateurs et améliorer fortement les performances de reconnaissance.

Enfin, un des problèmes majeurs de la robotique telle qu'elle existe actuellement est l'absence de standards. Par example, en téléphonie, les standards sont les architectures Android et OS X. En vision par ordinateur, la bibliothèque la plus utilisée est OpenCV, ce qui a permis de démocratiser la réalité augmentée. Jusqu'à maintenant, la plupart des architectures robotiques possédaient leur propre architecture logicielle et il était très délicat de proposer un scénario générique fonctionnant sur tous les robots. Avec l'architecture ROS, une tendance semble apparaître avec une augmentation des nombres de modules développés sous celle-ci. Cependant, malgré la multiplication des packages, aucun scénario fonctionnel qui pourrait servir de base d'implémentation n'est disponible. Il faudrait, en accord avec des sociologues, définir les grandes étapes d'un scénario le plus général possible, qui pourrait ensuite être décliné et spécifié suivant les besoins.

Ainsi la robotique d'assistance n'en est pour l'instant qu'à ses balbutiements, chaque scénario étant spécialisé sur une tâche très précise. Dans nos travaux, nous avons tenté de prendre en compte une partie du contexte et le comportement de l'utilisateur pour améliorer l'interaction. Nous espérons que le scénario que nous proposons inspirera de nouveaux travaux menants à la création du premier véritable robot d'assistance à domicile.

Abstract

This work is about human multimodal perception for human-robot interaction (HRI). This work was financed by the RIDDLE ANR Contint project (2012-2015). This project focuses on the development of an assisting robot for the elderly who experience small losses of memory. This project aims at coping with a growing need in human care for elder people living alone. Indeed in France, the population is aging and around 33% of the estimated population will be more than 60 years old by 2060. The goal is therefore to program an interactive robot (with perceptive capabilities), which would be able to learn the relationship between the user and a set of selected objects in their shared environment. In this field, lots of problems remain in terms of : (i) shared human-environment perception, (ii) integration on a robotic platform, and (iii) the validation of some scenarii about usual objects that involve both the robot and the elderly. The aim is to see the robot answer the user's interrogations about ten objects (defined by a preliminary study) with appropriate actions. For example, the robot will indicate the position of an object by moving towards it, grapping it or giving oral indications if it is not reachable. The RIDDLE project was formed by a consortium, with Magellium, the gerontology center of Toulouse, the MINC team from the LAAS-CNRS laboratory and Aldebaran Robotics. The final demonstrations will be led on the Roméo platform. This thesis has been co-directed by Frédéric Lerasle and Isabelle Ferrané, respectively from the RAP team of LAAS-CNRS and the SAMoVA team of IRIT.

Along the project, in partnership with the gerontology center, a robot scenario was determined following three major steps. During the first one -the "Monitoring step"- the robot is far from the user and waits for an intention of interaction. A "Proximal interaction step" is reached when the robot interacts with the user from a close position. Finally, the last step : the "Transition" allows the robot to move to reach the two previous ones. This scenario was built in order to create a not-intrusive proactive robot. This non-intrusiveness is materialized by the "monitoring step". The proactivity is achieved by the creation of a detector of user intention, allowing the robot to understand non-verbal information about the user's will to communicate with it.

The scientific contributions of this thesis include various aspects : robotic scenarii, the detector of user intention, a filtering technique based on particle swarm optimization algorithm, and finally a Baysian scheme built to improve the word error rate given distance information.

This thesis is divided in four chapters. The first one is about the detector of user intention. The second chapter moves on to the filtering technique. The third chapter will focus on the proximal interaction and the employed techniques, and finally the last chapter will deal with the robotic implementations.

Table des figures

Liste des tableaux

Bibliographie

Aizerman, M. A., Braverman, E. A., and Rozonoer, L. (1964). Theoretical foundations of the potential function method in pattern recognition learning. In *Automation and Remote Control*.

Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L., and Stent, A. (2000). An architecture for a generic dialogue shell. *Nat. Lang. Eng.*

Andrieu, C., Davy, M., and Doucet, A. (2001). Improved auxiliary particle filtering : applications to time-varying spectral analysis. In *Statistical Signal Processing, 2001. Proceedings of the 11th IEEE Signal Processing Workshop on*.

Bascetta, L., Ferretti, G., Rocco, P., Ardo, H., Bruyninckx, H., Demeester, E., and Di Lello, E. (2011). Towards safe human-robot interaction in robotic cells : An approach based on visual tracking and intention estimation. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*.

Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. (2008). Speeded-up robust features (surf). *Computer Vision and Image Understanding*.

Bogert, B., Healy, M., and Tukey, J. (1963). The quefrency alanysis of time series for echoes : Cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking. In *Proc. Symp. on Time Series Analysis*, pages 209–243.

Boudet, B., Fortin, C., Giacobini, T., Mollaret, C., Ferrané, I., Lerasle, F., and Rumeau, P. (2013). Etude pilote des objets recherchés sur une population de 60 personnes âgées ambulatoires - poster (33èmes journées annuelles de la société française de gériatrie et gérontologie, Paris, 08/10/2013-10/10/2013).

Boudet, B., Giacobini, T., Ferrané, I., Fortin, C., Mollaret, C., Lerasle, F., and Rumeau, P. (2014). Quels sont les objets égarés à domicile par les personnes âgées fragiles ? Une étude pilote sur 60 personnes. *Neurologie - Psychiatrie - Gériatrie*.

Brochard, R., Burger, B., Herbulot, A., and Lerasle, F. (2009). Measuring gaze orientation for human-robot interaction. In *Int. Workshop during IEEE Int. Symp. on Robot and Human Interactive Communication (RO-MAN'09), Toyama, Japan*.

Buss, M., Carton, D., Gonsior, B., Kuehnlenz, K., Landsiedel, C., Mitsou, N., de Nijs, R., Zlotowski, J., Sosnowski, S., Strasser, E., Tscheligi, M., Weiss, A., and Wollherr, D. (2011). Towards proactive human-robot interaction in human environments. In *Cognitive Infocommunications (CogInfoCom), 2011 2nd International Conference on*.

Can, B. and Artuner, H. (2013). A syllable-based turkish speech recognition system by using time delay neural networks (tdnns). In *Soft Computing and Pattern Recognition (SoCPaR), 2013 International Conference of*.

Changjiang, Y., Duraiswami, R., and Davis, L. (2005). Fast multiple object tracking via a hierarchical particle filter. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*.

Chen, S. Y. (2012). Kalman filter for robot vision : A survey. *Industrial Electronics*.

Chen, Y. and Medioni, G. (1991). Object modeling by registration of multiple range images. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*.

Chen-Chien, H. and Guo-Tang, D. (2012). Multiple object tracking using particle swarm optimization. *World Academy of Science, Engineering and Technology*.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*.

Ching-Han, C. and Miao-Chun, Y. (2011). Pso-based multiple people tracking. In *Digital Information and Communication Technology and Its Applications*.

Choi, J.-H. and Chang, J.-H. (2012). On using spectral gradient in conditional map criterion for robust voice activity detection. In *Network Infrastructure and Digital Content (IC-NIDC), 2012 3rd IEEE International Conference on*.

Clair, A. B. S., Mead, R., and Matarić, M. J. (2010). Monitoring and guiding user attention and intention in human-robot interaction. In *2010 IEEE International Conference on Robotics and Automation Workshop on Interactive Communication for Autonomous Intelligent Robots*.

Deléglise, P., Estève, Y., Meignier, S., and Merlin, T. (2005). The lium speech transcription system : a cmu sphinx iii-based system for french broadcast news. In *Interspeech*.

Deng, L., Li, J., Huang, J.-T., Yao, K., Yu, D., Seide, F., Seltzer, M., Zweig, G., He, X., Williams, J., Gong, Y., and Acero, A. (2013). Recent advances in deep learning for speech research at microsoft. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*.

Doucet, A. and Gordon, N. (1999). Efficient particle filters for tracking manoeuvring targets in clutter. In *Target Tracking : Algorithms and Applications, IEE Colloquium*.

Dov, D., Talmon, R., and Cohen, I. (2015). Audio-visual voice activity detection using diffusion maps. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*.

Fanelli, G., Dantone, M., Gall, J., Fossati, A., and Van Gool, L. (2013). Random forests for real time 3d face analysis. *Int. J. Comput. Vision*.

Fanelli, G., Gall, J., and Van Gool, L. (2011). Real time head pose estimation with random regression forests. In *Computer Vision and Pattern Recognition (CVPR)*.

Fleury, S., Herrb, M., and Chatila, R. (1997). Genom : A tool for the specification and the implementation of operating modules in a distributed robot architecture. In *In International Conference on Intelligent Robots and Systems*.

Fook, C., Hariharan, M., Yaacob, S., and Adom, A. (2012). A review : Malay speech recognition and audio visual speech recognition. In *Biomedical Engineering (ICoBE), 2012 International Conference on*.

Galliano, S., Geoffrois, E., Gravier, G., f. Bonastre, J., Mostefa, D., and Choukri, K. (2006). Corpus description of the ester evaluation campaign for the rich transcription of french broadcast news. In *In Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC 2006*.

Giremus, A., Doucet, A., Calmettes, V., and Tourneret, J.-Y. (2004). A rao-blackwellized particle filter for ins/gps integration. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*.

Gorur, O. and Erkmen, A. (2014). Elastic networks in reshaping human intentions by proactive social robot moves. In *Robot and Human Interactive Communication, 2014 RO-MAN : The 23rd IEEE International Symposium on*.

Han, J.-H., Kim, D.-H., and Kim, J.-W. (2009). Physical learning activities with a teaching assistant robot in elementary school music class. In *INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference on*.

Herranz, L., Xu, R., and Jiang, S. (2015). A probabilistic model for food image recognition in restaurants. In *Multimedia and Expo (ICME), 2015 IEEE International Conference on*.

Huang, J., Xuhui, S., and Wechsler, H. (1998). Face pose discrimination using support vector machines (svm). In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*.

Huber, B. (2013). Foot position as indicator of spatial interest at public displays. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*.

Hudlicka, E. and McNeese, M. D. (2002). Assessment of user affective and belief states for interface adaptation : Application to an air force pilot task. *User Modeling and User-Adapted Interaction*.

Huggins-daines, D., Kumar, M., Chan, A., Black, A. W., Ravishankar, M., and Rudnicky, A. I. (2006). Pocketsphinx : A free, real-time continuous speech recognition system for hand-held devices. In *in Proceedings of ICASSP*.

Isard, M. and Blake, A. (1998). Condensation—conditional density propagation for visual tracking. *International Journal of Computer Vision*.

Johal, W., Adam, C., Fiorino, H., Pesty, S., Jost, C., and Duhaut, D. (2014). Acceptability of a companion robot for children in daily life situations. In *Cognitive Infocommunications (CogInfoCom), 2014 5th IEEE Conference on*.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*.

Kondo, Y., Takemura, K., Takamatsu, J., and Ogasawara, T. (2012). Planning body gesture of android for multi-person human-robot interaction. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*.

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential Imputations and Bayesian Missing Data Problems. *Journal of the American Statistical Association*.

Kruijff, G.-J. M., Zender, H., Jensfelt, P., and Christensen, H. I. (2006). Clarification dialogues in human-augmented mapping. In *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-robot Interaction*.

Kuan, J.-Y., Huang, T.-H., and Huang, H.-P. (2010). Human intention estimation method for a new compliant rehabilitation and assistive robot. In *SICE Annual Conference 2010, Proceedings of*.

Kulić, E. A. and Croft, D. (2003). Estimating intent for human-robot interaction. In *International Conference on Advanced Robotics*.

Kumar, S., Rajasekar, P., Mandharasalam, T., and Vignesh, S. (2013). Handicapped assisting robot. In *Current Trends in Engineering and Technology (ICCTET), 2013 International Conference on*.

Lemaignan, S., Ros, R., Alami, R., and Beetz, M. (2011). What are you talking about ? Grounding dialogue in a perspective-aware robotic architecture. In *International Symposium in Robot and Human Interactive Communication*.

Lemaignan, S., Ros, R., Mösenlechner, L., Alami, R., and Beetz, M. (2010). Oro, a knowledge management module for cognitive architectures in robotics. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Li, T., Sun, S., and Sattar, T. (2013). Adapting sample size in particle filters through kld-resampling. *Electronics Letters*.

Li, Y., Li, Y., Hu, T., and Lv, Z. (2015). An automatic semantic web service composition method based on ontology. In *Computer and Information Science (ICIS), 2015 IEEE/ACIS 14th International Conference on*.

Maganti, H., Gatica-Perez, D., and McCowan, I. (2007). Speech enhancement and recognition in meetings with an audio-visual sensor array. *Audio, Speech, and Language Processing, IEEE Transactions on*.

Mallet, A., Fleury, S., and Bruyninckx, H. (2002). A specification of generic robotics software components : future evolutions of genom in the orocos context. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*.

Martin, M., Van De Camp, F., and Stiefelhagen, R. (2014). Real time head model creation and head pose estimation on consumer depth cameras. In *3D Vision (3DV), 2014 2nd International Conference on*.

Mei-Ping, S. and Guo-chang, G. (2004). Research on particle swarm optimization : a review. In *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*.

Mollaret, C., Lerasle, F., Ferrané, I., and Pinquier, J. (2014). A particle swarm optimization inspired tracker applied to visual tracking. In *Image Processing (ICIP), 2014 IEEE International Conference on*.

Mollaret, C., Mekonnen, A., Ferrane, I., Pinquier, J., and Lerasle, F. (2015). Perceiving user's intention-for-interaction : A probabilistic multimodal data fusion scheme. In *Multimedia and Expo (ICME), 2015 IEEE International Conference on*.

Murphy-Chutorian, E. and Trivedi, M. (2009). Head pose estimation in computer vision : A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pages 607–626.

Onuma, Y., Kamado, N., Saruwatari, H., and Shikano, K. (2012). Real-time semi-blind speech extraction with speaker direction tracking on kinect. In *Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*.

Ooko, R., Ishii, R., and Nakano, Y. (2011). Estimating a user's conversational engagement based on head pose information. In *Intelligent Virtual Agents*. Springer Berlin Heidelberg.

OpenKinect (2010). libfreenect.

Padeleris, P., Zabulis, X., and Argyros, A. (2012). Head pose estimation on depth data based on particle swarm optimization. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*.

Pandey, A., Ali, M., Warnier, M., and Alami, R. (2011). Towards multi-state visuo-spatial reasoning based proactive human-robot interaction. In *Advanced Robotics (ICAR), 2011 15th International Conference on*.

Pellegrini, T., Guyot, P., Angles, B., Mollaret, C., and Mangou, C. (2014). Towards soundpainting gesture recognition (regular paper). In *Audio Mostly, Aalborg, Denmark, 01/10/2014-03/10/2014*.

Pinheiro, M., Bicho, E., and Erlhagen, W. (2010). A dynamic neural field architecture for a proactive assistant robot. In *Biomedical Robotics and Biomechatronics (BioRob), 2010 3rd IEEE RAS and EMBS International Conference on*.

Poli, R. (2008). Analysis of the publications on the applications of particle swarm optimisation. In *Journal of Artificial Evolution and Applications*.

PrimeSense (2010). *OpenNI User Guide*. OpenNI organization.

Qiao, T. and Dai, S.-L. (2013). Fast head pose estimation using depth data. In *Image and Signal Processing (CISP), 2013 6th International Congress on*.

Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition*.

Rios-Martinez, J., Escobedo, A., Spalanzani, A., and Laugier, C. (2012). Intention driven human aware navigation for assisted mobility. In *Workshop on Assistance and Service robotics in a human environment at IROS*.

Rocha, R., Freire, V., and Alencar, M. (2014). Voice segmentation system based on energy estimation. In *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*.

Schmid, A., Weede, O., and Worn, H. (2007). Proactive robot task selection given a human intention estimate. In *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*.

Seemann, E., Nickel, K., and Stiefelhagen, R. (2004). Head pose estimation using stereo vision for human-robot interaction. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*.

Seltzer, M. and Stern, R. (2006). Subband likelihood-maximizing beamforming for speech recognition in reverberant environments. *Audio, Speech, and Language Processing, IEEE Transactions on*.

Sha, F., Bae, C., Liu, G., Zhao, X., Chung, Y. Y., and Yeh, W. (2015). A categorized particle swarm optimization for object tracking. In *Evolutionary Computation (CEC), 2015 IEEE Congress on*.

Simon, L. S. and Vincent, E. (2015). *Combining blockwise and multi-coefficient stepwise approches in a general framework for online audio source separation*. PhD thesis.

Spiliotopoulos, D., Androutsopoulos, I., and Spyropoulos, C. D. (2001). Human-robot interaction based on spoken natural language dialogue. In *in : Proceedings of the European Workshop on Service and Humanoid Robots*.

Tavakkoli, A., Kelley, R., King, C., Nicolescu, M., Nicolescu, M., and Bebis, G. (2007). A vision-based architecture for intent recognition. In *Advances in Visual Computing*.

Trawicki, M., Johnson, M., Ji, A., and Osiejuk, T. (2012). Multichannel speech recognition using distributed microphone signal fusion strategies. In *Audio, Language and Image Processing (ICALIP), 2012 International Conference on*.

Valenti, R., Sebe, N., and Gevers, T. (2012). Combining head pose and eye location information for gaze estimation. *Image Processing, IEEE Transactions on*.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*.

Wan, E. and Merwe, R. V. D. (2000). The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*.

Wan, Y.-L., Zhang, T.-Q., Wang, Z.-C., and Jin, J. (2013). Robust speech recognition based on multi-band spectral subtraction. In *Image and Signal Processing (CISP), 2013 6th International Congress on*.

Wang, Y., Huang, S., and Wei, Y. (2013). A voice activity detection algorithm with sub-band detection based on time-frequency characteristics of mandarin. In *Image and Signal Processing (CISP), 2013 6th International Congress on*.

Webb, J. and Ashley, J. (2012). *Beginning Kinect Programming with the Microsoft Kinect SDK*.

Wei, X., Zhang, X., and Wang, Y. (2012). Research on a detection and recognition method of tactile-slip sensation used to control the elderly-assistant amp ; walking-assistant robot. In *Automation Science and Engineering (CASE), 2012 IEEE International Conference on*.

Wu, J. and Etzkorn, L. (2015). Validation of an approach for finding good anchor nodes in ontologies in the semantic web. In *SoutheastCon 2015*.

Xiao, Y., Zhang, Z., Beck, A., Yuan, J., and Thalmann, D. (2014). Human-robot interaction by understanding upper body gestures. *Presence*.

Xiong, G., Gong, J., Zhuang, T., Zhao, T., Liu, D., and Chen, X. (2007). Development of assistant robot with standing-up devices for paraplegic patients and elderly people. In *Complex Medical Engineering, 2007. CME 2007. IEEE/ICME International Conference on*.

Yamazaki, K., Ueda, R., Nozawa, S., Kojima, M., Okada, K., Matsumoto, K., Ishikawa, M., Shimoyama, I., and Inaba, M. (2012). Home-assistant robot for an aging society. *Proceedings of the IEEE*.

Young, S., Gasic, M., Thomson, B., and Williams, J. (2013). Pomdp-based statistical spoken dialogue systems : a review. *Proceedings of the IEEE*.

Zhang, J., Feng, Y., Ning, G., and Ji, F. (2015). Noise adaptive stream fusion based on feature component rejection for robust multi-stream speech recognition. In *Advanced Computational Intelligence (ICACI), 2015 Seventh International Conference on*.

Zhang, X., Hu, W., Li, W., Qu, W., and Maybank, S. (2009). Multi-object tracking via species based particle swarm optimization. In *ICCV Workshops*.

Zhang, X., Hu, W., Maybank, S., Li, X., and Zhu, M. (2008). Sequential particle swarm optimization for visual tracking. In *CVPR*.

Résumé de thèse

Cette thèse porte sur la perception multimodale de l'homme pour l'Interaction Homme-Robot (IHR). Elle a été financée par le projet ANR Contint RIDDLE (2012 – 2015). Ce projet est centré sur le développement d'un robot d'assistance pour les personnes âgées atteintes de troubles cognitifs légers. Celui-ci a pour but de répondre à un besoin grandissant d'aide à domicile envers les personnes âgées vivant seules. En effet, la population vieillissant de plus en plus, on estime qu'environ 33% des français auront plus de 60 ans en 2060. L'enjeu est donc de programmer un robot interactif (via ses capacités perceptuelles) capable d'apprendre la relation entre l'usager et un sous-ensemble d'objets du quotidien de ce dernier, soit des objets pertinents, présents ou possiblement égarés dans l'espace partagé du robot et de l'usager. Dans ce cadre, il reste de nombreux verrous à lever, notamment en termes de : (i) perception conjointe de l'homme et de son environnement, (ii) d'intégration sur un système robotisé, (iii) de validation par des scénarii mettant en jeu le robot et une personne âgée en interaction avec quelques objets usuels. La finalité du projet est de voir le robot répondre aux interrogations relatives à une dizaine d'objets courants (définis par une étude préliminaire sur une population qui se plaint de troubles cognitifs) par des actions appropriées. Par exemple, le robot signalera l'emplacement d'un objet en se déplaçant vers lui, en le saisissant ou en donnant des indications orales quant à sa position si l'objet n'est pas atteignable. Le projet RIDDLE est multipartenaire : il regroupe la société Magellium, le Gérontopôle de Toulouse, l'équipe MINC du LAAS-CNRS et l'entreprise Aldebaran Robotics dont le robot doit servir de plateforme pour les démonstrations finales. Cette thèse a été co-encadrée par Frédéric Lerasle et Isabelle Ferrané respectivement enseignants-chercheurs dans les équipes RAP du LAAS-CNRS et SAMoVA de l'IRIT-UPS.

Lors de ce projet, nous avons, en partenariat avec le gérontopôle, défini un scénario robotique regroupant trois phases principales. Une phase de *monitoring* où le robot se trouve loin de l'utilisateur et l'observe de sa position, en attente d'une demande d'interaction, une phase d'interaction proximale où le robot se trouve proche de l'utilisateur et interagit avec lui, et enfin la transition qui permet au robot de passer d'une phase à l'autre. Ce scénario est donc construit de manière à créer un robot d'interaction proactif mais non-intrusif. Le caractère non-intrusif est matérialisé par la phase de *monitoring*. La proactivité est, quant à elle, matérialisée par la création d'un détecteur d'intentionnalité permettant au robot de comprendre de manière non-verbale la volonté de l'utilisateur de communiquer avec lui.

Les contributions scientifiques de cette thèse recoupent divers aspects du projet : le scénario robotique, le détecteur d'intentionnalité, une technique de filtrage par essaim de particules, et enfin une technique bayésienne d'amélioration du taux d'erreur de mot à partir d'informations de distance.

Cette thèse est divisée en quatre chapitres. Le premier traite du détecteur d'intentionnalité, le deuxième de la technique de filtrage développée, le troisième de la phase d'interaction proximale et des techniques employées, et enfin le dernier chapitre est centré sur les implémentations robotiques.